

Other Information:	N/A
Assumptions:	N/A

Table 56 - <Mail Service> Send E-mail

### 3. Functional Requirements

#### 3.1. Conceptual Diagram

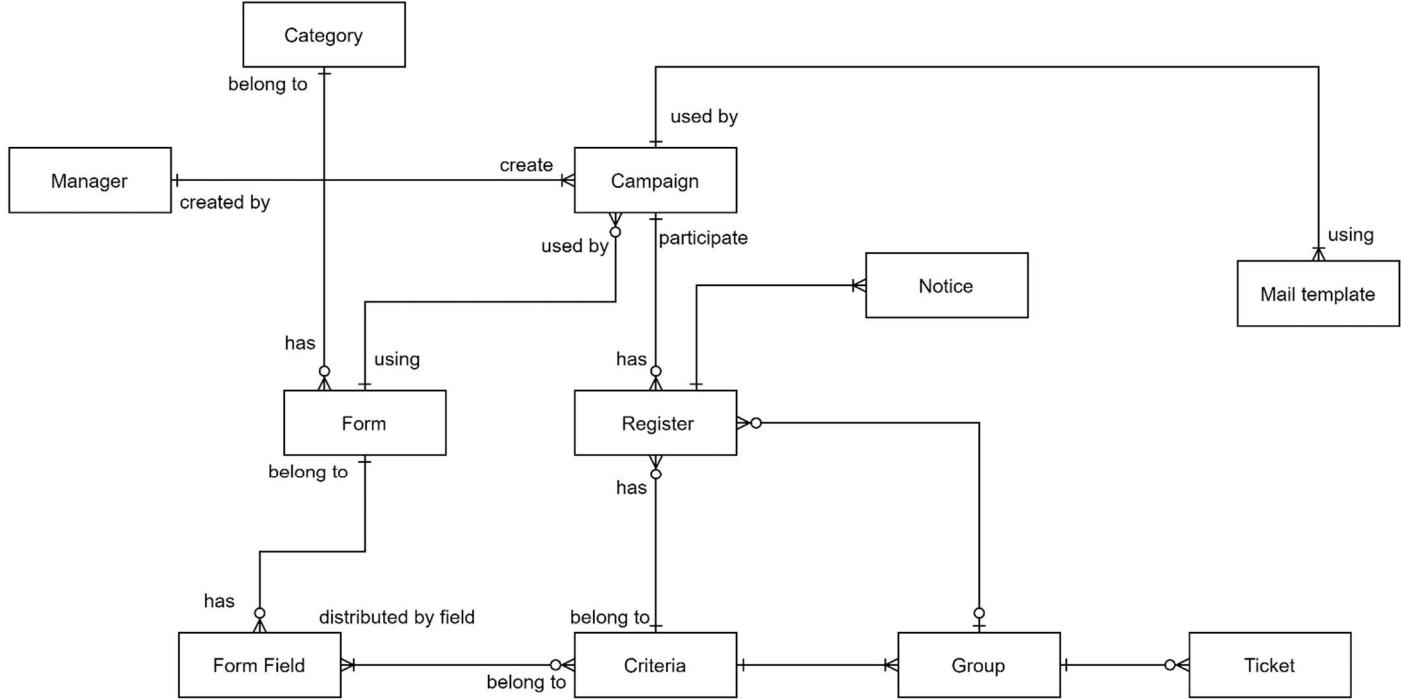


Figure 59 - Conceptual Diagram

### 3.2. Entity Relationship Diagram

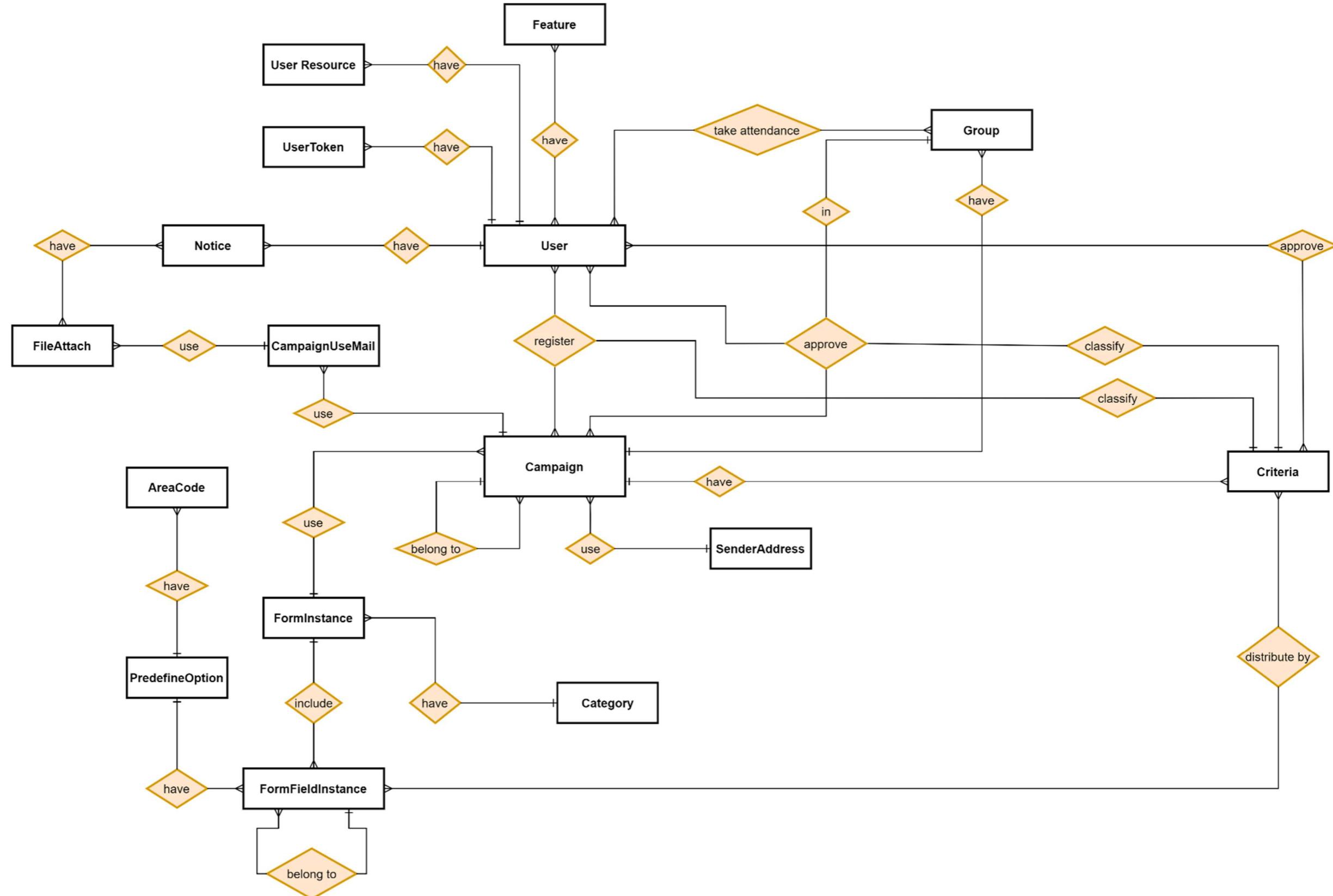


Figure 60 - Entity Relationship Diagram (Layer 1)

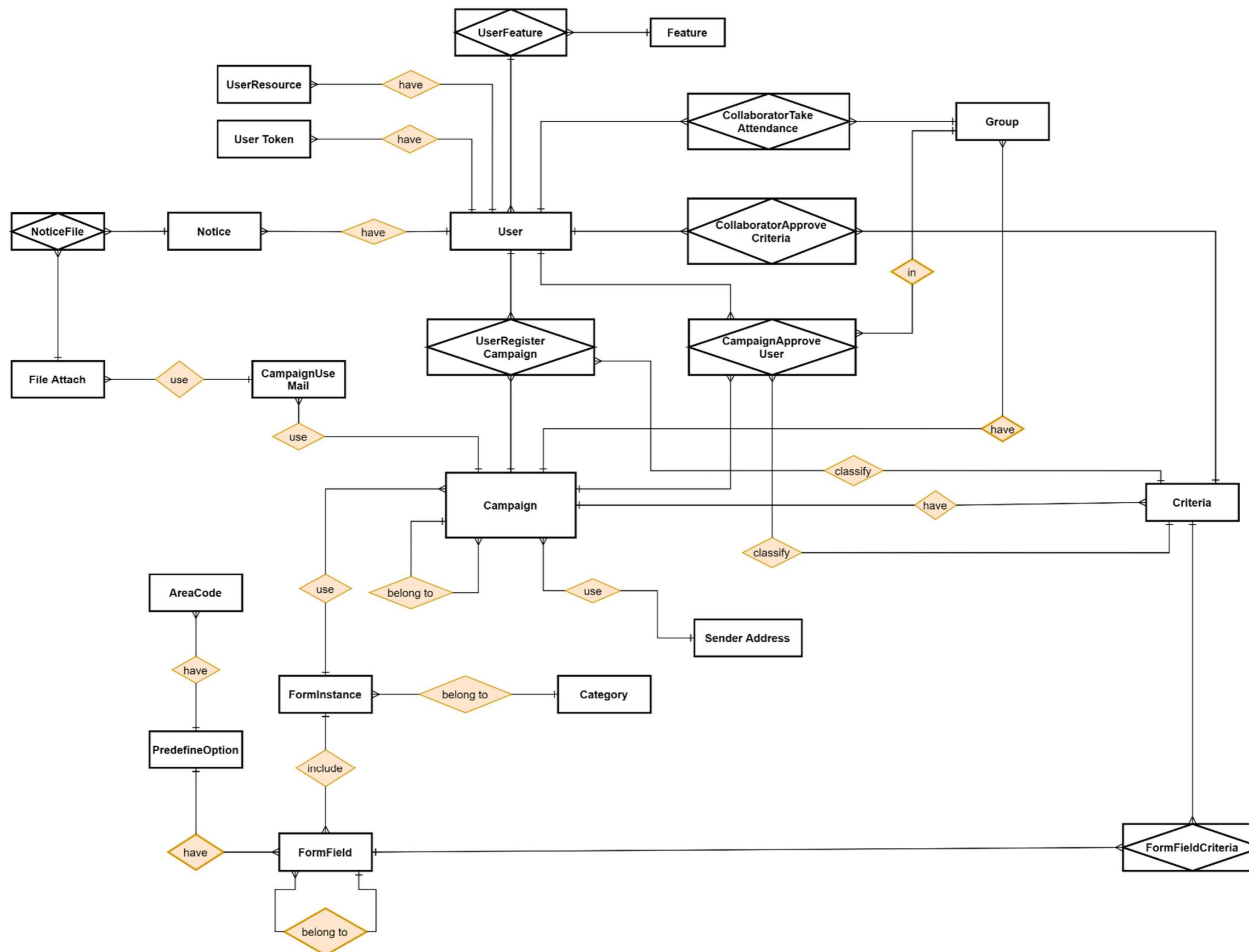
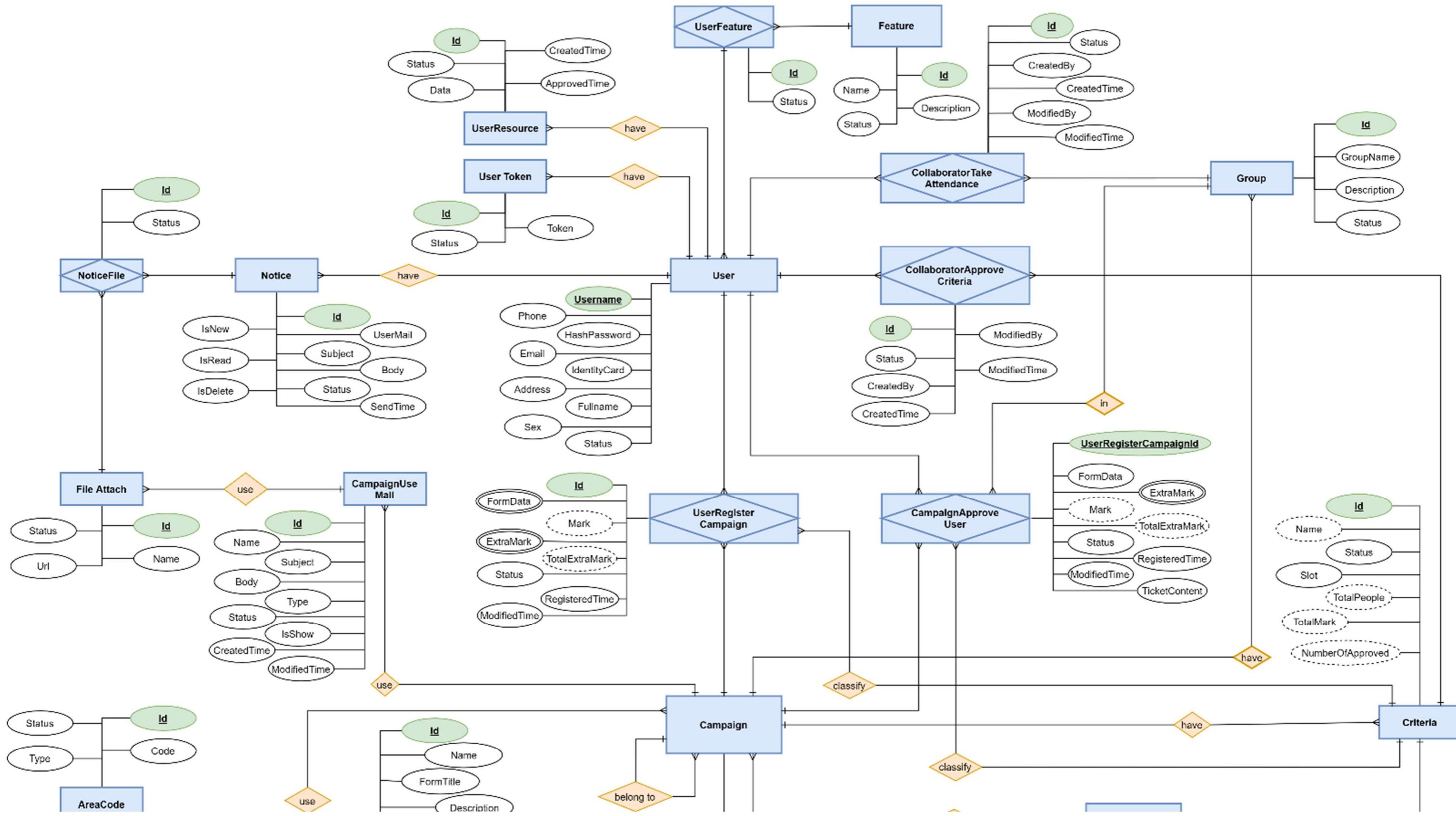


Figure 61 - Entity Relationship Diagram (Layer 2)



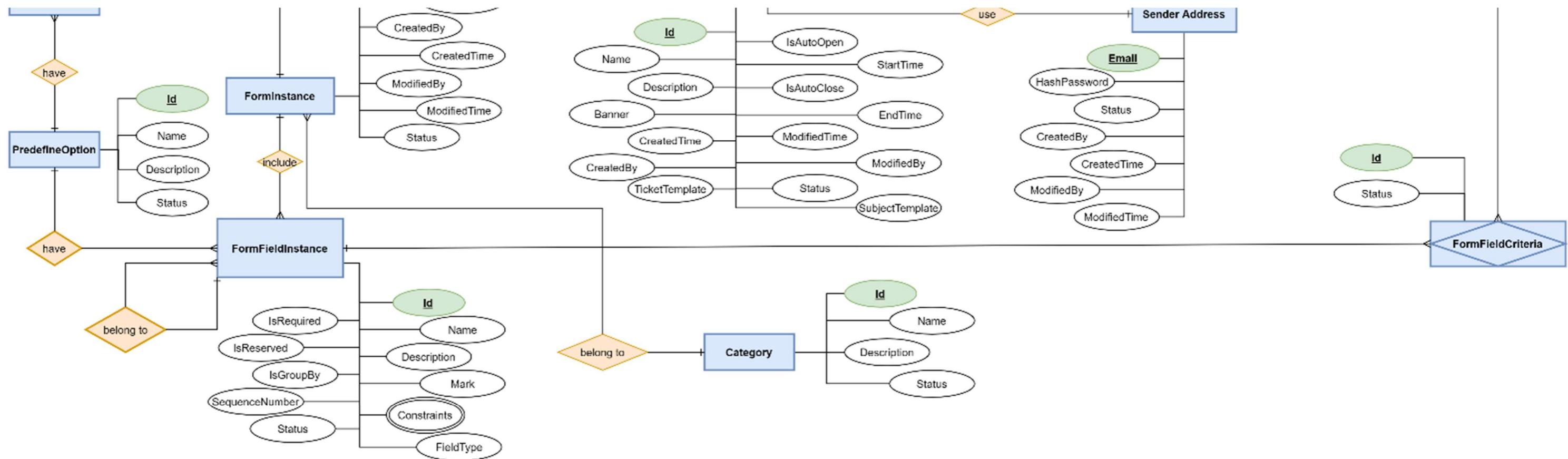


Figure 62 - Entity Relationship Diagram (with attributes)

No.	Entity	Description
1	User	<p>Contain the information of user          Dùng để chứa các tt như          Dùng để chức năng abc          --          Thông tin của user là 1 thực thể để chứa thông tin ...          Từ thông tin này thì ...</p>
2	UserToken	Contain the token information of user
3	UserResource	Contain the resource information of user
4	UserFeature	Contain the feature information of user
5	Feature	Contain the information of feature
6	Notice	Contain the information of mail have been sent to user
7	NoticeFile	Contain the information of file have been attached in mail which have been sent to user
8	FileAttach	Contain the information of file attached in mail
9	UserRegisterCampaign	Contain the information of the registrant, include: form data registration, registration time,...
10	Campaign	Contain the information of campaign
11	CampaignUseMail	Contain the information of mail template which used by campaign
12	FormInstance	Contain the information of form
13	FormFieldInstance	Contain the information of form field in form
14	PredefineOption	Contain the information of option field in form
15	AreaCode	Contain the information of area
16	Category	Contain the information of category
17	SenderAddress	Contain the information of mail address which used as a mail default of campaign to send mail notification user
18	CampaignApproveUser	Contain the information of user who have been approved in specific campaign
19	Criteria	Contain the information of criteria. Criteria have lots of user who have been distributed, base on their answer from form registration
20	FormFieldCriteria	Contain the information of form field used to distribute user into criteria
21	Group	Contain the information of group which created form many criteria
22	CollaboratorApproveCriteria	Contain the information of task include: which criteria that collaborators assigned to approve

23	CollaboratorTakeAttendance	Contain the information of task include: which group that collaborators assigned to take attendance
----	----------------------------	---

*Table 57 - Entity Data Dictionary*

## 4. Non-Functional Requirements

### 4.1. External Interfaces

#### 4.1.1. User Interfaces

- The UI shall support Vietnamese.
- The registrants' application must be fully showed in mobile web browsers.

#### 4.1.2. Software Interfaces

- OCR Detection Service: The system can intergrated with FPT.AI Service.

#### 4.1.3. Communication Interfaces

- Authentication Service: The system can be intergrated with Firebase Authentication.
- File Storage Service: The system can be intergrated with Firebase Storage.
- The system can send e-mail to registrants to notify the registration's status.
- The system can use HTTP Protocol to transfer data.

## 4.2. Quality Attributes

### 4.2.1. Usability

- All texts, labels and messages should be written in Vietnamese.
- The web application for the registrant should be simple and easy to use without training.
- The web application for the manager should require no more than one day of training to be used.

### 4.2.2. Reliability

- The system should handle 500 requests per a day.

### 4.2.3. Portability

- The manager's software and registrant's software are web-based applications.
- The applications are working well in Chrome Browser version 65 and higher.

#### **4.2.4. Security**

- YCR system is currently divided into 3 roles: Registrant, Collaborator and Manager.
- All input data are validated before saving to database.
- All privacy information such as password is encrypted to ensure security.
- Each role only accesses a group of functions.
- The system always checks for authorization and authentication before doing any features.

### **IV. Software Design Description**

#### **1. Overall Description**

##### **1.1. Assumptions**

The system is designed basing on these following assumptions:

- Windows 10
- Chrome 65
- SQL Server 2014
- End-user: SAC manager and every student in Ho Chi Minh city

##### **1.2. Design Constraints**

The system should be compiled with following items:

- End-user environment:
  - SAC manager: in Windows 10 with Chrome browser version 65 or higher.
  - Students: in Chrome browser version 65.
- Security requirement: prevent user from cheating as much as possible.
- Network communication: this system depends on network connection.

#### **2. System Architecture Design**

##### **2.1. Overall Architecture**

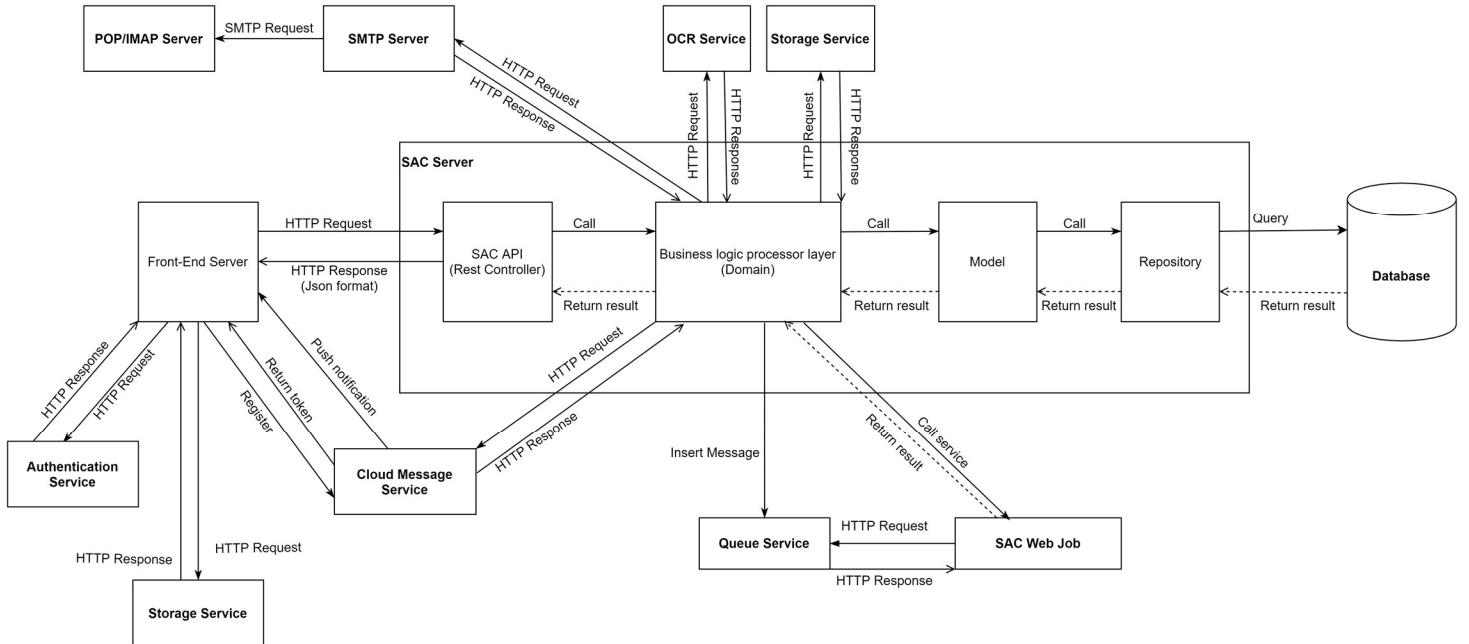


Figure 63 - System Overall Architecture Diagram

No.	Name	Description
1	Front-End Server	Client component to render and display data from server. Client component provides functions: <ul style="list-style-type: none"> <li>- For manager and collaborator: manage campaign, manage form, approve registrants, publish ticket and take attendance.</li> <li>- For registrant: register campaign, receive approval's result and receive ticket.</li> </ul>
2	Authentication Service	This service is in charge of authenticating 3 <sup>rd</sup> party account of the registrant and send result to our system.
3	Storage Service	Storage service to store resources of registrants and managers. The resources are: identity images and form images when register provides to the form.
4	Cloud Message Service	Service to notify message from the system to users when the registrants have registered a campaign, updated form information, canceled registration or when manager notifies approval, publishes ticket to registrants.
5	SAC API	It handles request from Front End and response from the system.
6	Business logic process layer (Domain)	It handles business logic of the system.

<b>7</b>	Model	It supports the system to handle data.
<b>8</b>	Repository	It handles reading and writing data in the system.
<b>9</b>	Database	It stores data of the system.
<b>10</b>	Queue Service	It stores messages that the web job needs to execute.
<b>11</b>	SAC Web Job	It handles automation jobs such as open or close campaign on time as it has scheduled before.
<b>12</b>	OCR Service	It detects and transfers identity images of registrants to text.
<b>13</b>	SMTP Service	It handles sending emails.
<b>14</b>	POP/IMAP Service	It receives and shows emails of the system to clients.

Table 58 - Architecture Diagram Dictionary

## 2.2. Component Diagram

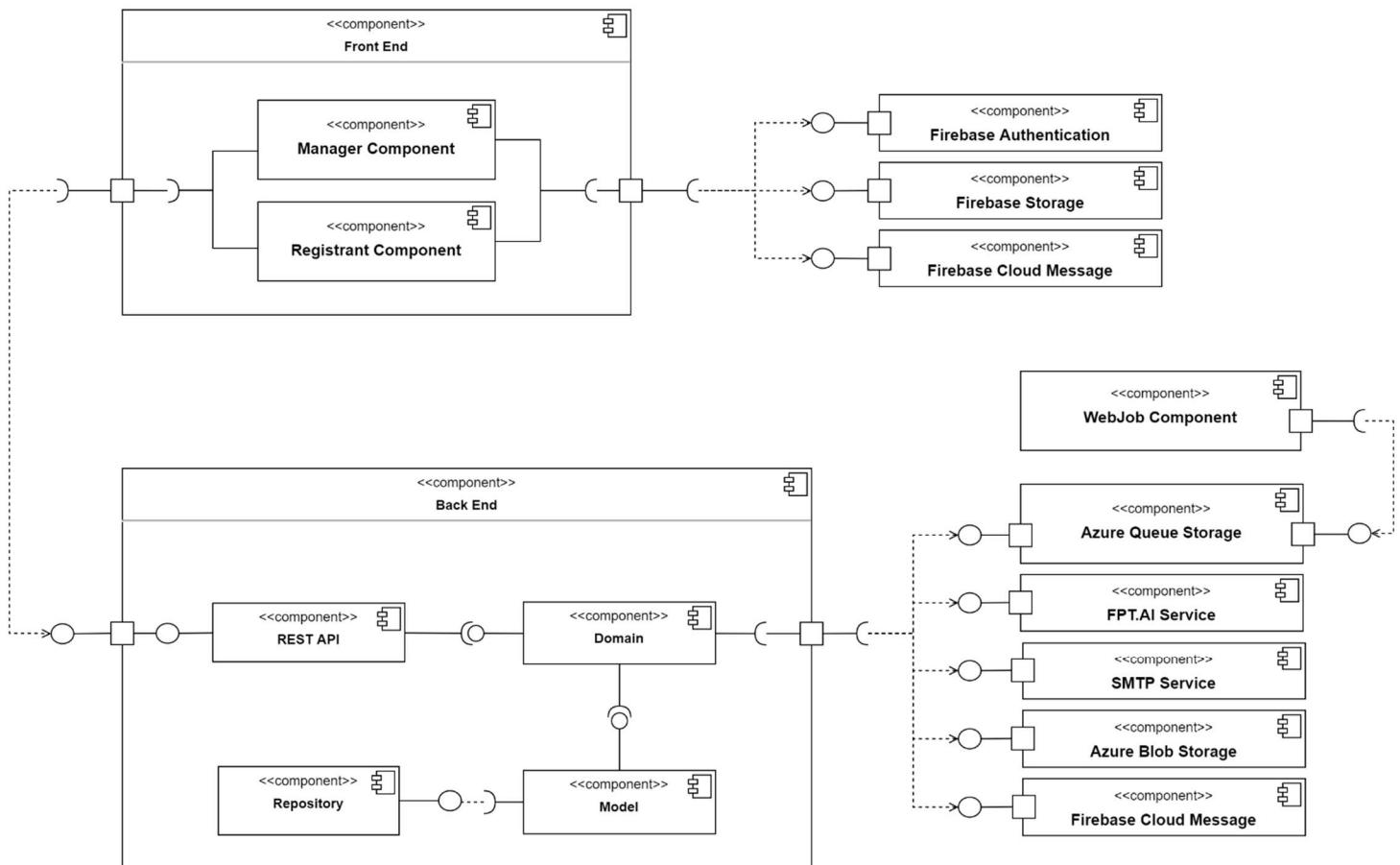


Figure 64 - Component Diagram

No.	Name	Description
-----	------	-------------

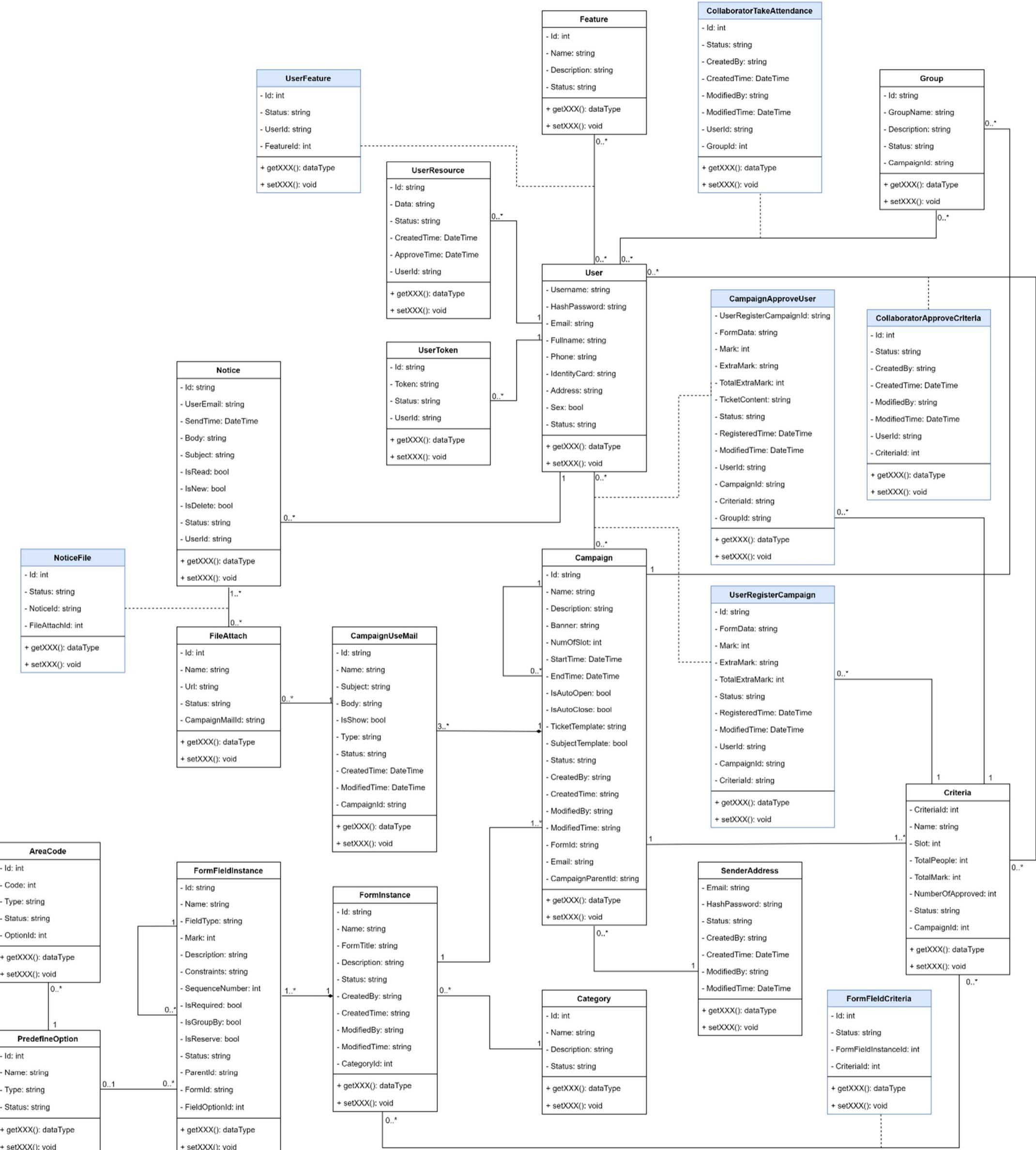
<b>1</b>	Front End	Client component to render and display data from server. Client component provides functions: <ul style="list-style-type: none"><li>- For manager and collaborator: manage campaign, manage form, approve registrants, publish ticket and take attendance.</li><li>- For registrant: register campaign, receive approval's result and receive ticket.</li></ul>
<b>2</b>	Manager Component	It provides features: manage campaign, manage form, approve registrants, publish ticket and take attendance.
<b>3</b>	Registrant Component	It provides features: register campaign, receive approval's result and receive ticket.
<b>4</b>	Firebase Authentication	Registrants use gmail account to be authenticated in YCR system. This component is in charge of authenticating Gmail account of the registrant and send result to our system.
<b>5</b>	Firebase Storage	Storage service to store resources of registrants and managers. The resources are: identity images and form images when register provides to the form.
<b>6</b>	Firebase Cloud Message	Service to notify message from the system to users when the registrants have registered a campaign, updated form information, canceled registration or when manager notifies approval, publishes ticket to registrants.
<b>7</b>	Back End	It receives HTTP Request from Front End, process business logic, store data to database and response data result to Front End.
<b>8</b>	REST API	It handles request from Front End and response from the system.
<b>9</b>	Domain	It handles business logic of the system.
<b>10</b>	Repository	It handles reading and writting data in the system.
<b>11</b>	Model	It supports the system to handle data.
<b>12</b>	WebJob Component	It handles automation jobs such as open or close campaign on time as it has scheduled before.
<b>13</b>	Azure Queue Storage	It stores messages that the web job needs to execute.
<b>14</b>	FPT.AI Service	It detects and transfers identity images of registrants to text.
<b>15</b>	SMTP Service	It handles sending emails.
<b>16</b>	Azure Blob Storage	Storage service to store QR code images, which are generated in the process of backend.

Table 59 - Component Diagram Dictionary

### **3. System Detailed Design**

#### **3.1. Class Diagram**

### 3.1.1. Class Diagram



*Figure 65 - Class Diagram*

### 3.1.2. Class Specifications

#### User class

No.	Attribute	Data Type	Visibility	Description
1	Username	string	private	Unique identifier of a user
2	HashPassword	string	private	The hash password of account
3	Email	string	private	The email of a user
4	Fullscreen	string	private	The full name of a user
5	Phone	string	private	The phone number of a user
6	IdentityCard	string	private	The identity card of a user
7	Address	string	private	The address of a user
8	Sex	bool	private	The gender of a user
9	Status	string	private	The status of a user

Figure 66 - <Class Specification> User

#### UserResource class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of a user resource
2	Data	string	private	The data of resource
3	Status	string	private	The status of resource
4	CreatedTime	DateTime	private	The created time of resource
5	ApprovedTime	DateTime	Private	The time that the resource has been approved.
6	UserId	string	private	Unique identifier of an user have this resource

Figure 67 - <Class Specification> UserResource

#### UserToken class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of a token
2	Token	string	private	Token string for push notification
3	Status	string	private	Status of the token
4	UserId	string	private	Unique identifier of a user has this token

Figure 68 - <Class Specifications> UserToken

### UserFeature class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier
2	Status	string	private	Unique identifier of a user has this feature
3	UserId	string	private	Unique identifier of a feature
4	FeatureId	string	private	The full name of a user

Figure 69 - <Class Specifications> UserFeature

### Feature class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of a feature
2	Name	string	private	The name of feature
3	Description	string	private	The description of feature
4	Status	string	private	The status of this feature

Figure 70 - <Class Specifications> Feature

### Notice class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of notice
2	UserEmail	string	private	Email of user receive this notice
3	SendTime	DateTime	private	The time when notice created
4	Body	string	private	The body of notice
5	Subject	string	private	The subject of notice
6	IsRead	bool	private	The flag indicates that notice is read or not.
7	IsNew	bool	private	The flag indicates that notice is new or not.
8	IsDelete	bool	private	The flag indicates that notice is deleted or not.
9	Status	string	private	The status of the notice
10	UserId	string	private	Unique identifier of user receives this notice

Figure 71 - <Class Specifications> Notice

### NoticeFile class

No.	Attribute	Data Type	Visibility	Description
1	Id	int	private	Unique identifier
2	Status	string	private	The status of file attached in this notice
3	NoticeId	string	private	Unique identifier of a notice which using this file attach
4	FileAttachId	int	private	Unique identifier of a file attach which used by notice

Figure 72 - <Class Specifications> NoticeFile

### FileAttach class

No.	Attribute	Data Type	Visibility	Description
1	Id	int	private	Unique identifier of a file attach
2	Name	string	private	The name of a file attach
3	Url	string	private	The URL of file attach
4	Status	string	private	The status of file attach
5	CampaignMailId	string	private	Unique identifier of mail template using this file attach

Figure 73 - <Class Specifications> FileAttach

### Campaign class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of a campaign
2	Name	string	private	The name of a campaign
3	Description	string	private	The description of campaign
4	Banner	string	private	The banner url of campaign
5	NumOfSlot	int	private	The number slot of campaign
6	StartTime	DateTime	private	The start time of campaign
7	EndTime	DateTime	private	The end time of campaign
8	IsAutoOpen	bool	private	Indicate the campaign open automatically or not
9	IsAutoClose	bool	private	Indicate the campaign close automatically or not
10	TicketTemplate	string	private	The ticket template of campaign

11	SubjectTemplate	string	private	The subject template of campaign
12	Status	string	private	The status of the campaign
13	CreatedBy	string	private	The username of user who creates this campaign
14	CreatedTime	string	private	The create time of campaign
15	ModifiedBy	string	private	The username of user who modifies this campaign
16	ModifiedTime	string	private	The modified time of campaign
17	FormId	string	private	The Unique identifier of form which campaign using
18	Email	string	private	The mail address used for send notice to registrant
19	Campaign ParentId	string	private	The unique identifier of parent campaign

Figure 74 - <Class Specifications> Campaign

### CampaignUseMail class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of a mail template
2	Name	string	private	The name of a mail template
3	Subject	string	private	The subject of mail template
4	Body	string	private	The body of mail template
5	IsShow	bool	private	Indicate that mail template is show or not
6	Type	string	private	The type of mail template. There are 3 type: Update, Register, Cancel
7	Status	string	private	The status of mail template
8	CreatedTime	DateTime	private	The create time of mail template
9	ModifiedTime	DateTime	private	The modified time of mail template
10	CampaignId	string	private	Unique identifier of campaign using this mail template

Figure 75 - <Class Specifications> CampaignUseMail

### SenderAddress class

No.	Attribute	Data Type	Visibility	Description
1	Email	string	private	Unique identifier of email
2	HashPassword	string	private	The password hashed of email

3	Status	string	private	The status of email address
4	CreatedBy	string	private	The username of user who creates this email
5	CreatedTime	DateTime	private	The create time of email
6	ModifiedBy	string	private	The username of user who modifies this email
7	ModifiedTime	DateTime	private	The modified time of email

Figure 76 - <Class Specifications> SenderAddress

### FormInstance class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of a form instance
2	Name	string	private	The name of form instance
3	FormTitle	string	private	The title of form instance
4	Description	string	private	The description of form instance
5	Status	string	private	The status of form instance
6	CreatedBy	string	private	The username of the user who creates this form instance
7	CreatedTime	DateTime	private	The created time of this form instance
8	ModifiedBy	string	private	The username of the user who modifies this form instance
9	ModifiedTime	DateTime	private	The username of the user who modifies this form instance
10	CategoryId	int	private	Unique identifier of category which this form instance belong to

Figure 77 - <Class Specifications> FormInstance

### FormFieldInstance class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of a form field instance
2	Name	string	private	The name of form field instance
3	FieldType	string	private	The field type of form field instance. There are 6 type: text, textarea, option, select, radio, checkbox
4	Mark	int	private	The mark of form field instance

5	Description	string	private	The description of form field instance
6	Constraints	string	private	The constraints expression of form field instance
7	SequenceNumber	int	private	Indicate the sequence of form field instance
8	IsRequired	bool	private	Indicate form field instance require or not
9	IsGroupBy	bool	private	Indicate form field instance is used to group by
10	IsReserve	bool	private	Indicate form field instance use to process extra mark or not
11	Status	string	private	The status of form field instance
12	ParentId	string	private	Unique identifier of parent form field instance
13	FormId	string	private	Unique identifier of form that this field belong to
14	FieldOptionId	int	private	Unique identifier of field option id

Figure 78 - <Class Specifications> FormFieldInstance

### PredefineOption class

No.	Attribute	Data Type	Visibility	Description
1	Id	int	private	Unique identifier of a predefine option
2	Name	string	private	The name of predefine option
3	Type	string	private	The type of predefine option. There are 2 type: School, City
4	Status	string	private	The status of predefine option

Figure 79 - <Class Specifications> PredefineOption

### AreaCode class

No.	Attribute	Data Type	Visibility	Description
1	Id	int	private	Unique identifier of an area code
2	Code	string	private	The name of an area code
3	Type	string	private	The type of an area code. There are 2 type: CCCD, CMND
4	Status	string	private	The status of an area code
5	OptionId	int	private	Unique identifier of a predefine option

Figure 80 - <Class Specifications> AreaCode

### Category class

No.	Attribute	Data Type	Visibility	Description
1	Id	int	private	Unique identifier of a category
2	Name	string	private	The name of category
3	Description	string	private	The description of category
4	Status	string	private	The status of category

Figure 81 - <Class Specifications> Category

### UserRegisterCampaign class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of a registrant
2	FormData	string	private	The data of form registration
3	Mark	int	private	The mark of answer of question in form registrant
4	ExtraMark	string	private	The data of extra mark of registrant
5	TotalExtraMark	int	private	The extra mark of registrant
6	Status	string	private	The status of registrant
7	RegisteredTime	DateTime	private	The register time of registrant
8	ModifiedTime	DateTime	private	The time registrant modifies form registration
9	UserId	string	private	Unique identifier of user has registered
10	CampaignId	string	private	Unique identifier of campaign
11	CriteriaId	string	private	The id of criteria which registrant belong to

Figure 82 - <Class Specifications> UserRegisterCampaign

### Criteria class

No.	Attribute	Data Type	Visibility	Description
1	CriteriaId	int	private	Unique identifier of a criteria
2	Name	string	private	The name of criteria
3	Slot	int	private	The slot of criteria
4	TotalPeople	int	private	The number people of criteria

5	TotalMark	int	private	The total mark of people in criteria
6	NumberOfApproved	int	private	The number of registrants has been approved in criteria
7	Status	string	private	The status of criteria
8	CampaignId	int	private	Unique identifier of campaign

Figure 83 - <Class Specifications> Criteria

### FormFieldCriteria class

No.	Attribute	Data Type	Visibility	Description
1	Id	int	private	Unique identifier
2	Status	string	private	The status of this relation between FormFieldInstance and Criteria
3	FormFieldInstanceId	int	private	Unique identifier of form field instance
4	CriteriaId	int	private	Unique identifier of criteria

Figure 84 - <Class Specifications> FormFieldCriteria

### CampaignApproveUser class

No.	Attribute	Data Type	Visibility	Description
1	UserRegisterCampaignId	string	private	Unique identifier of a registrant
2	FormData	string	private	The data of form registration
3	Ma�t des	int	private	The mark of answer of question in form registrant
4	ExtraMark	string	private	The data of extra mark of registrant
5	TotalExtraMark	int	private	The extra mark of registrant
6	TicketContent	string	private	The identity card of a user
7	Status	string	private	The status of registrant
8	RegisteredTime	DateTime	private	The register time of registrant
9	ModifiedTime	DateTime	private	The time registrant modifies form registration
10	UserId	string	private	Unique identifier of user has registered
11	CampaignId	string	private	Unique identifier of campaign

12	Criteriald	string	private	The id of criteria which registrant belong to
13	GroupId	string	private	The id of group which registrant belong to

Figure 85 - <Class Specifications> CampaignApproveUser

### Group class

No.	Attribute	Data Type	Visibility	Description
1	Id	string	private	Unique identifier of a group
2	GroupName	string	private	The name of group
3	Description	string	private	The description of group
4	Status	string	private	The status of group
5	CampaignId	string	private	Unique identifier of a campaign which includes this group

Figure 86 - <Class Specifications> Group

### CollaboratorApproveCriteria class

No.	Attribute	Data Type	Visibility	Description
1	Id	int	private	Unique identifier
2	Status	string	private	The status of task for collaborator approves criteria
3	CreatedBy	string	private	The username of user assigns this task
4	CreatedTime	DateTime	private	The created time of this task
5	ModifiedBy	string	private	The username of user modified this task
6	ModifiedTime	DateTime	private	The modified time of this task
7	UserId	string	private	Unique identifier of user
8	Criteriald	int	private	Unique identifier of a criteria

Figure 87 - <Class Specifications> CollaboratorApproveCriteria

### CollaboratorTakeAttendance class

No.	Attribute	Data Type	Visibility	Description
1	Id	int	private	Unique identifier of a user
2	Status	string	private	The status of this task
3	CreatedBy	string	private	The username of user creates this task

4	CreatedTime	DateTime	private	The created time of this task
5	ModifiedBy	string	private	The username of user modifies this task
6	ModifiedTime	DateTime	private	The modified time of this task
7	UserId	string	private	Unique identifier of this task for collaborator take attendance
8	GroupId	int	private	Unique identifier of a group

*Figure 88 - <Class Specifications> CollaboratorTakeAttendance*

## **3.2. Sequence Diagram**

### **3.2.1. Create campaign**

*Summary:* This diagram shows the process for manager to provide campaign information, form information and config email. The system will receive those information and store data to database.

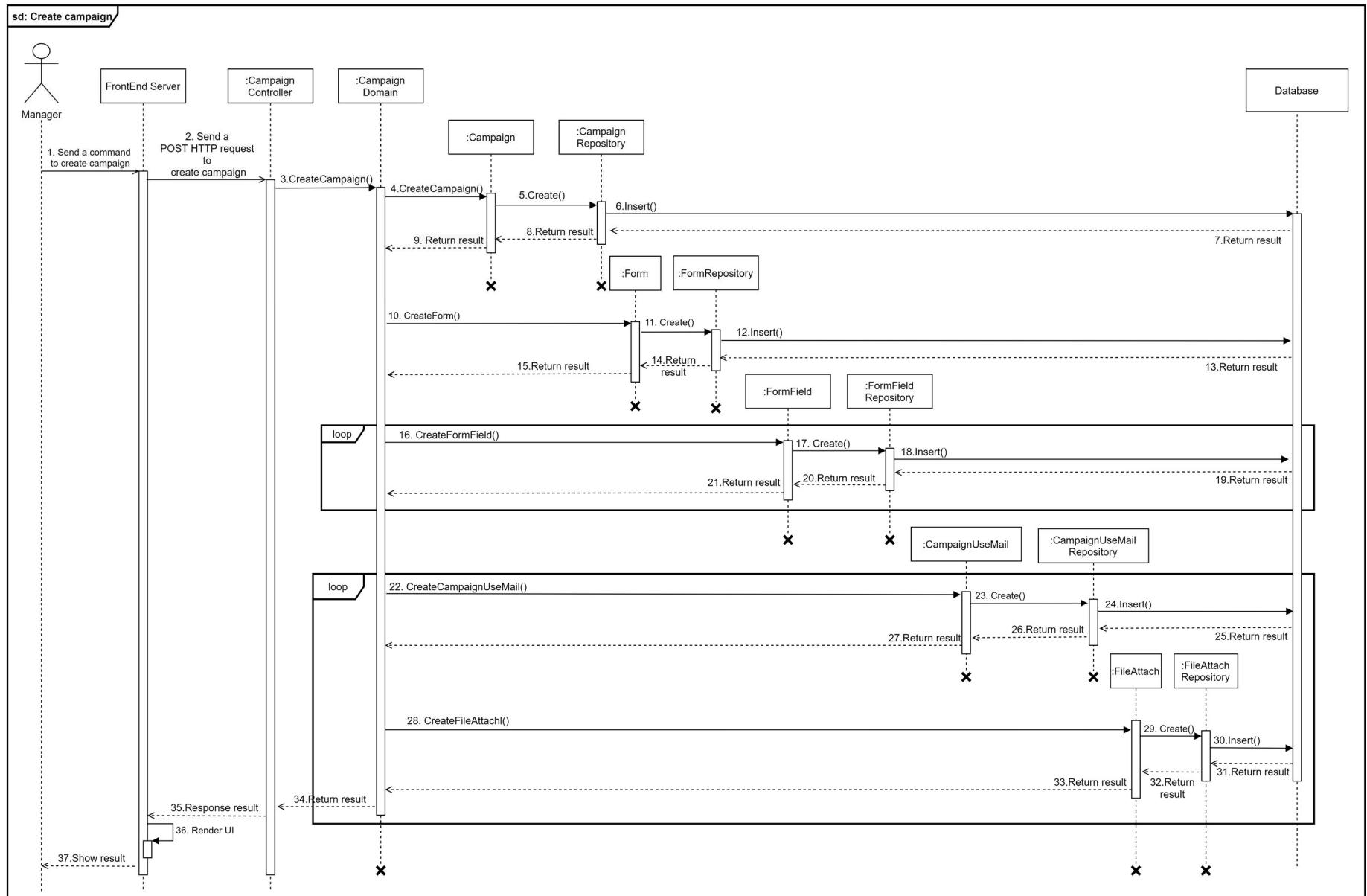


Figure 89 - <Sequence Diagram> Create campaign

### 3.2.2. Open or Close campaign automatically

*Summary:* The application allows the manager to config the time for opening or closing a campaign automatically on time. This diagram shows the process for system handler to automatically open or close a campaign.

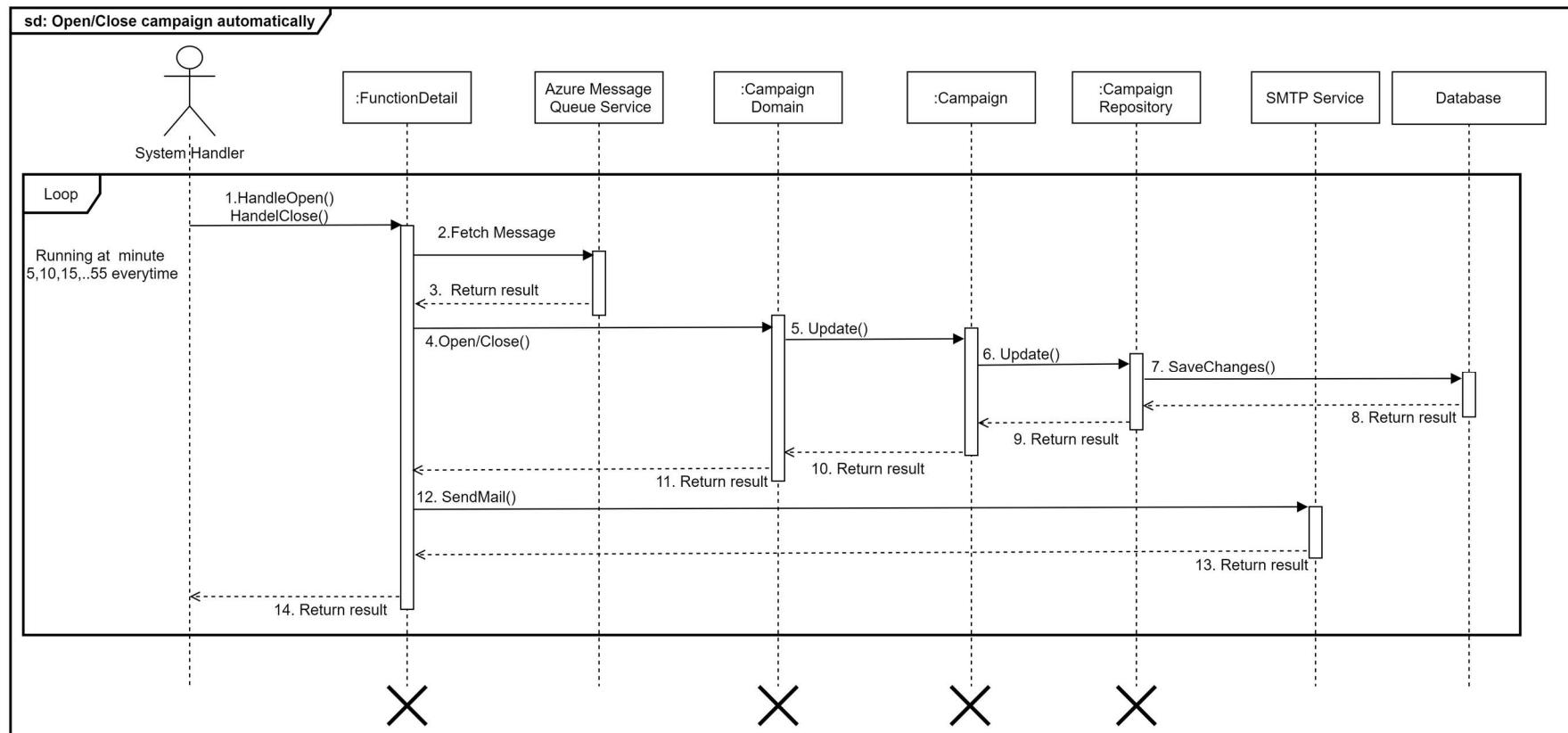
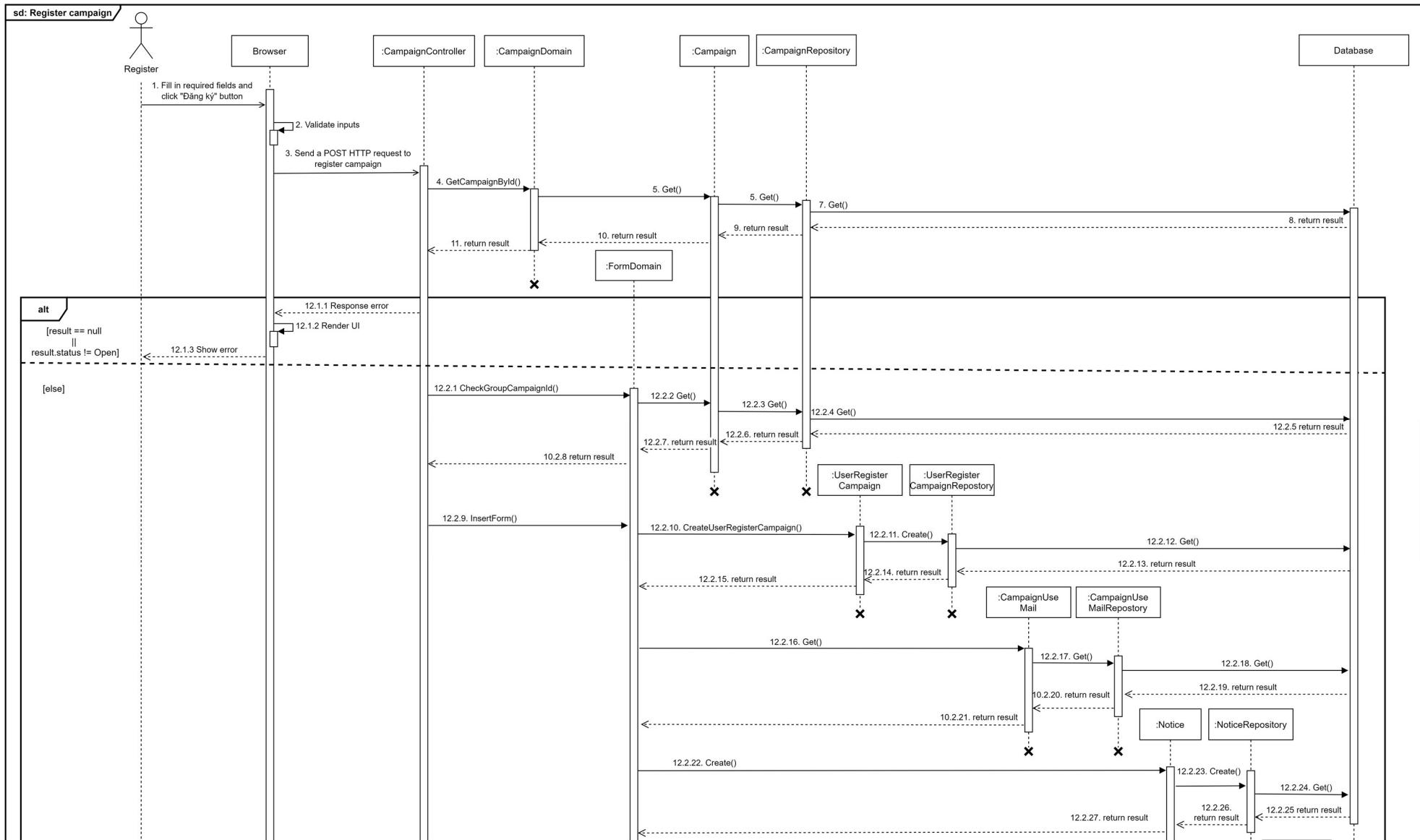


Figure 90 - <Sequence Diagram> Open or Close campaign automatically

### 3.2.3. Register campaign

*Summary:* This diagram shows the process of a registrant to register to a campaign. When registering to a campaign, the system will store form data of the registrant, get the email content and then send email to the registrant.



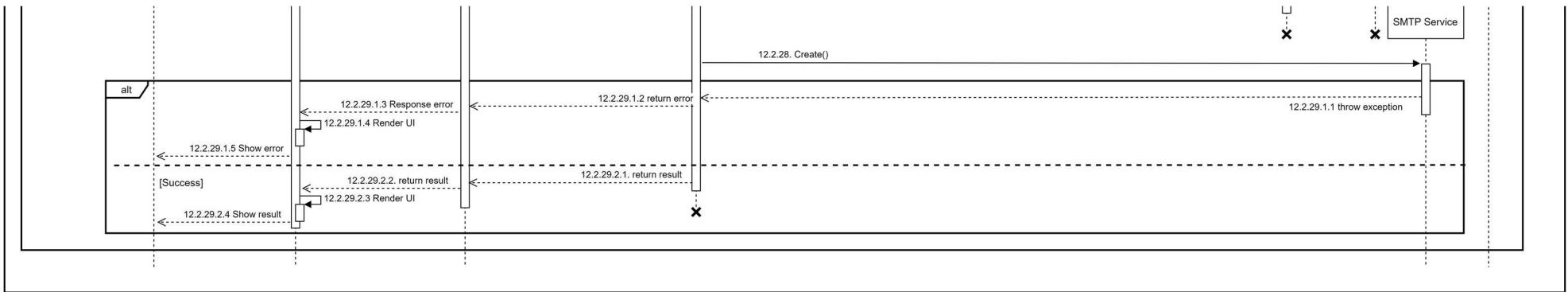
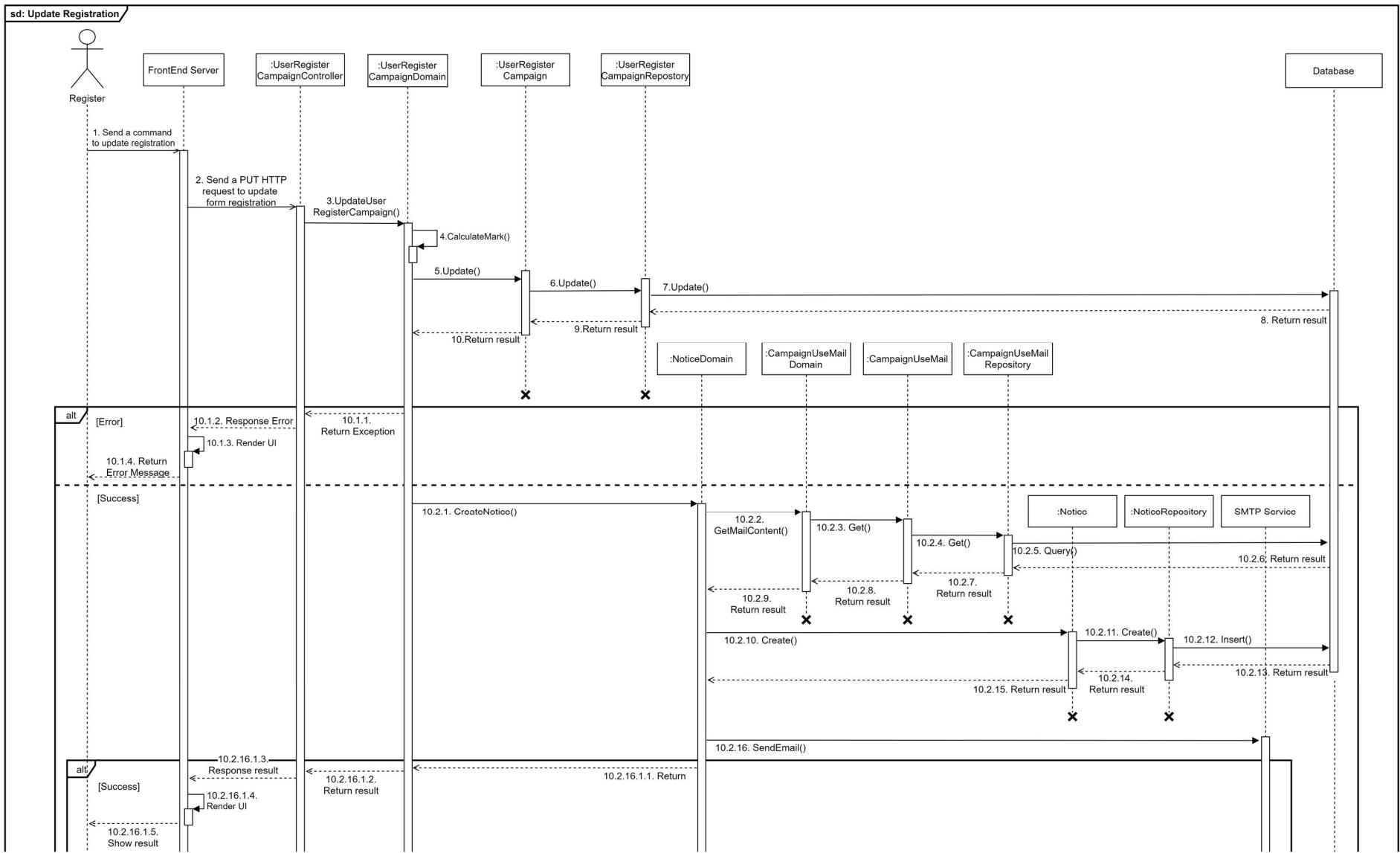


Figure 91 - <Sequence Diagram> Register campaign

### 3.2.4. Update registration

**Summary:** After complete registration, registrants may need to update some information in the form. This diagram shows the process of update registration form's information and send email to confirm with the registrant.



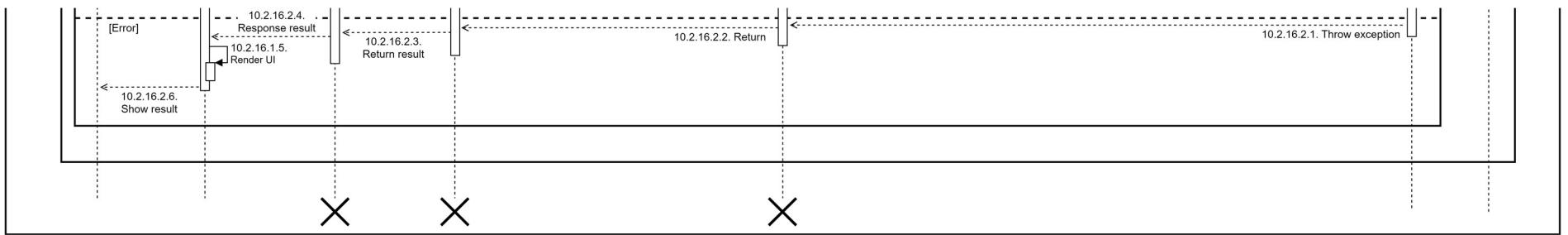
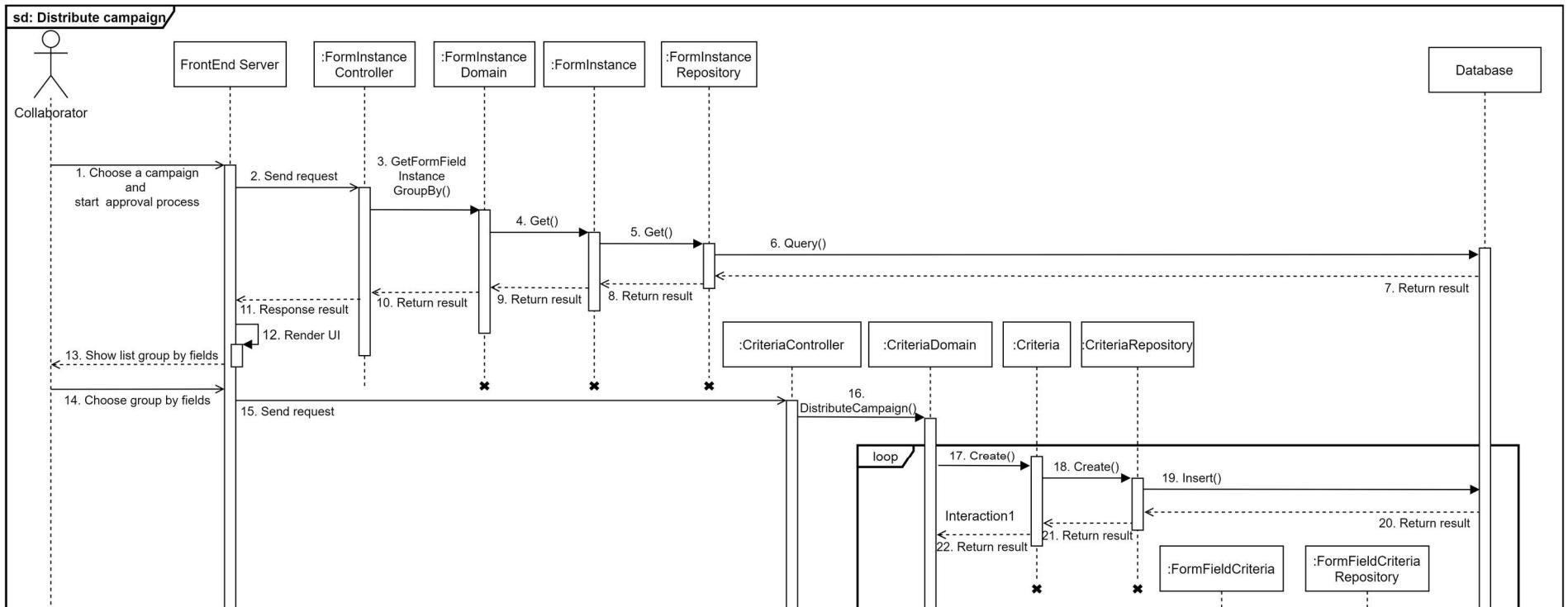


Figure 92 - <Sequence Diagram> Update registration

### 3.2.5. Distribute campaign

**Summary:** After close the campaign, manager needs to approve registrants. Approval process bases on some criteria. So, this diagram shows the process to distributes registrants list of a campaign to multiple groups for approval.



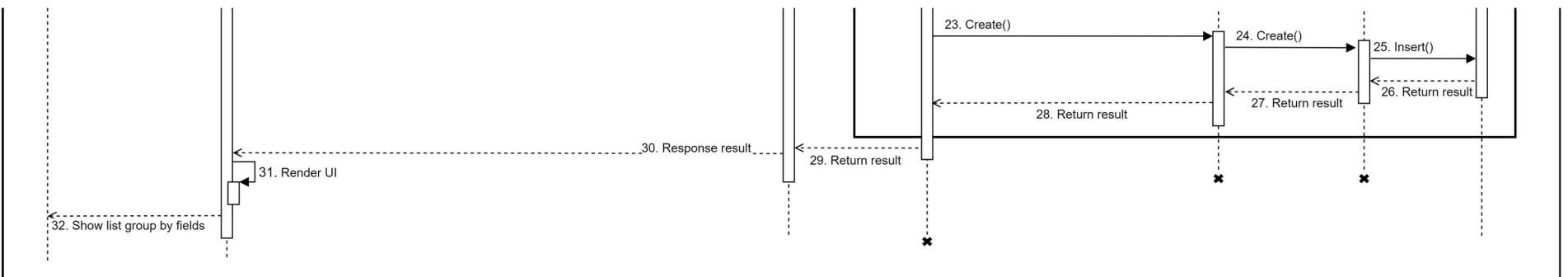
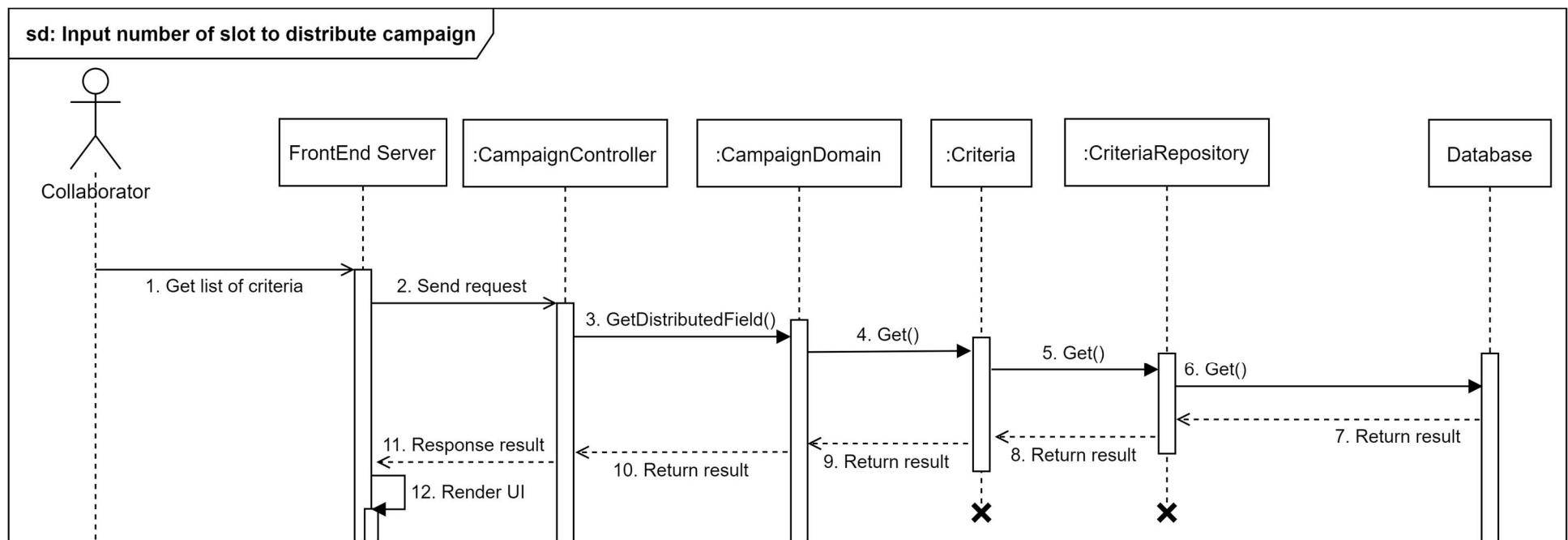


Figure 93 - <Sequence Diagram> Distribute campaign

### 3.3. Input number to criteria

**Summary:** After distributing camapign, manager starts approval process by inputting number of registrants that are available for the criteria. This diagram shows the process of the system receives quantity of criteria.



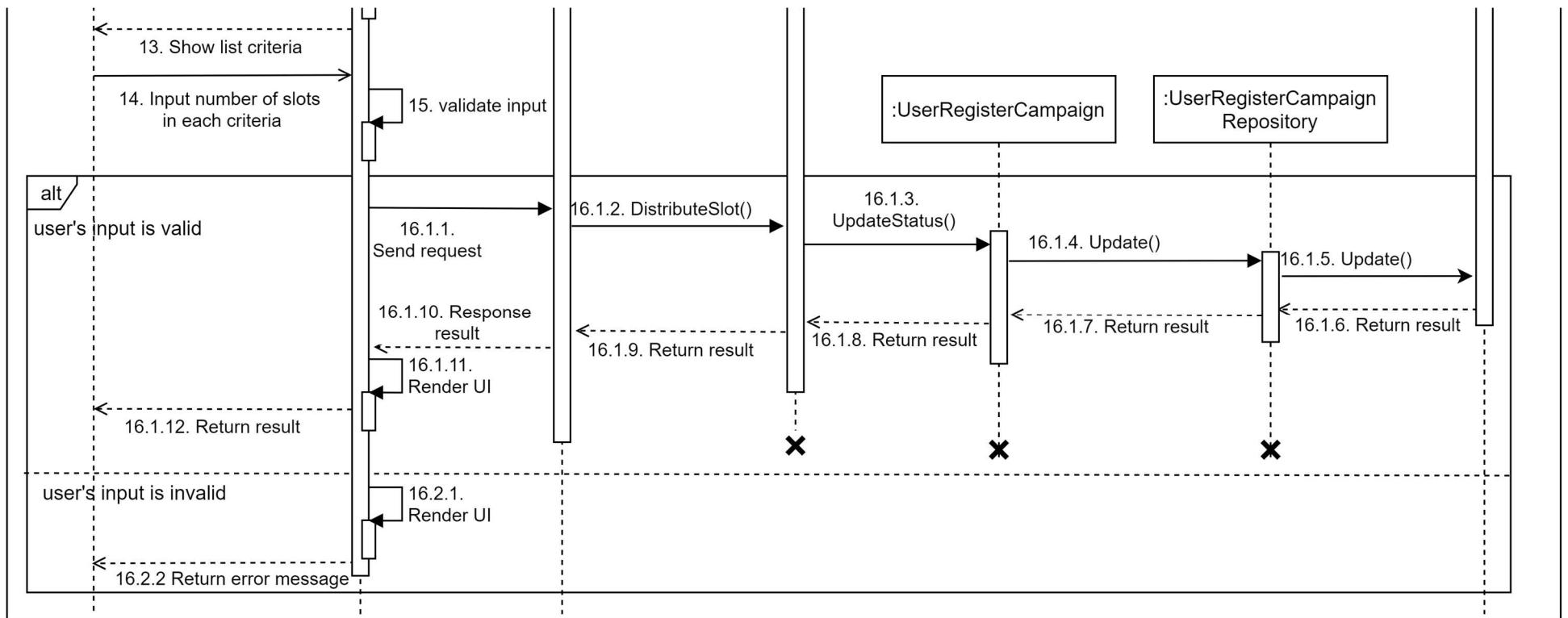


Figure 94 - <Sequence Diagram> Input number to criteria

### 3.4. Automatically rank registrants

**Summary:** After input available quantity for each criteria, the system will automatically approve registrants by ranking registrants in the criteria. This diagram shows the process to rank the registrants.

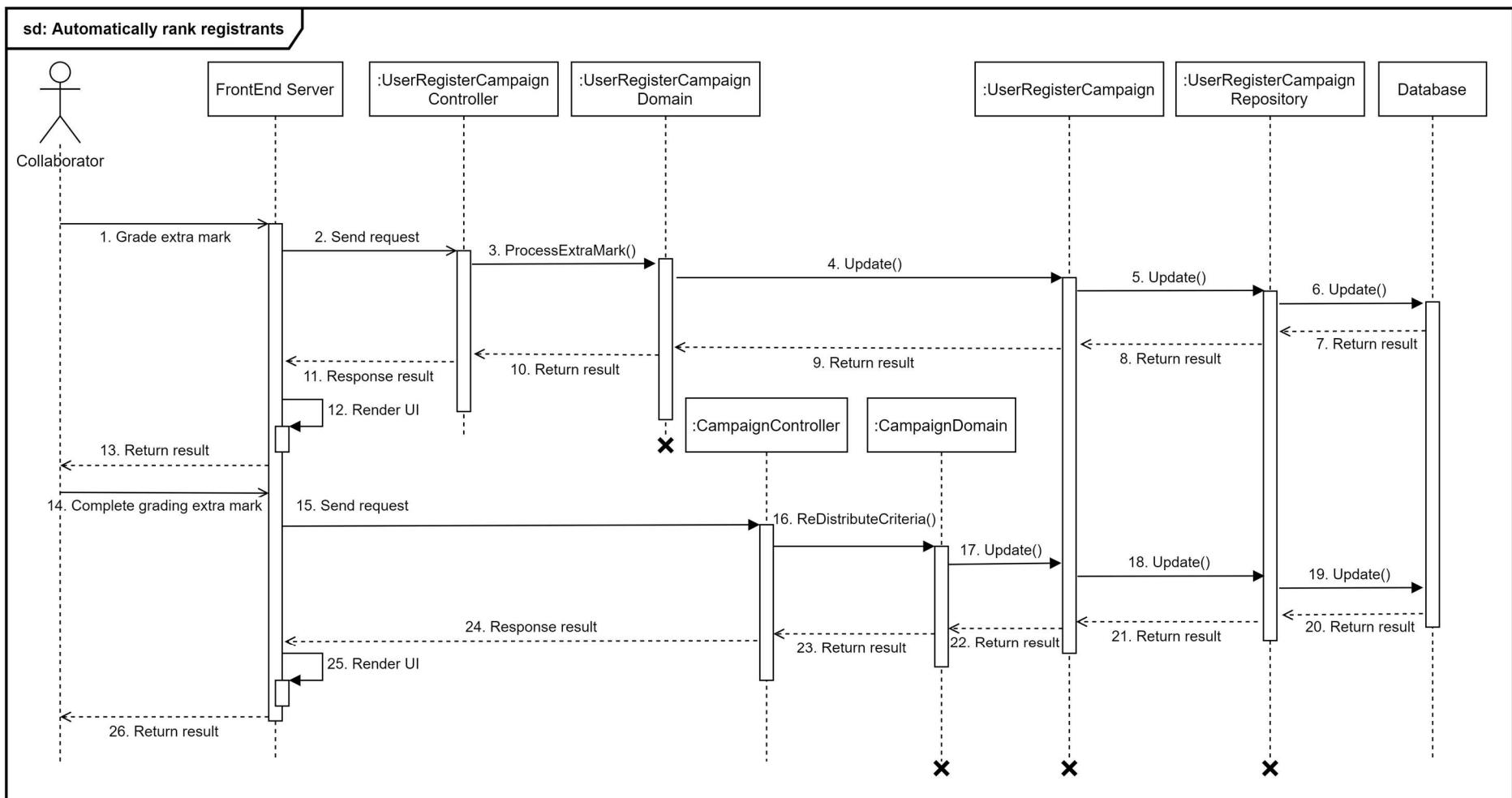


Figure 95 - <Sequence Diagram> Automatically rank registrants

### 3.5. Distribute registrants into groups

**Summary:** Ticket is one of the process in our system. To create ticket, managers need create groups, to add participants have the same ticket's information. This diagram shows the process to distribute registrants into groups.

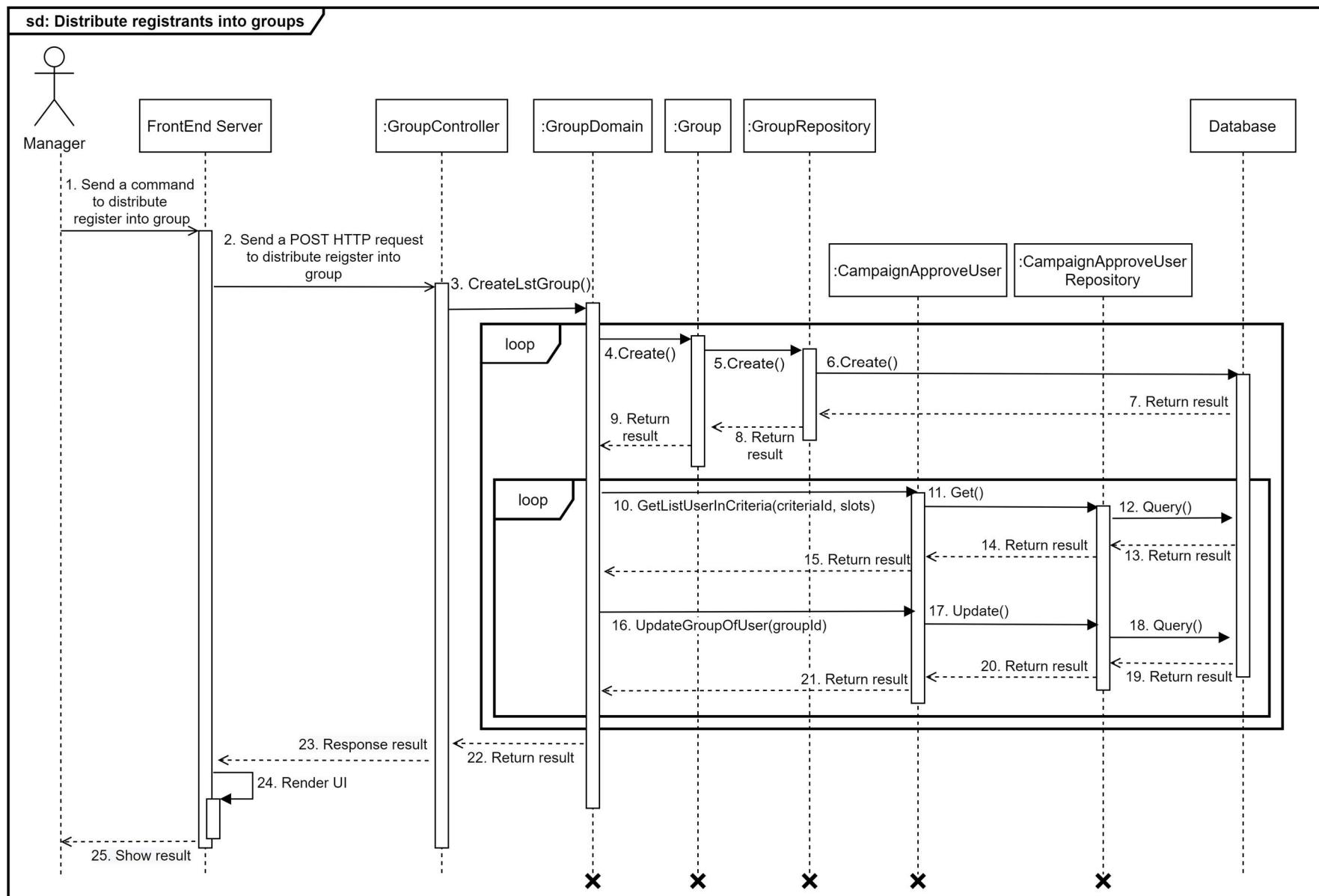


Figure 96 - <Sequence Diagram> Distribute registrants into groups

### 3.6. State Diagram

### 3.7. Campaign Status

*Summary:* This diagram shows every state of a campaign in our system.

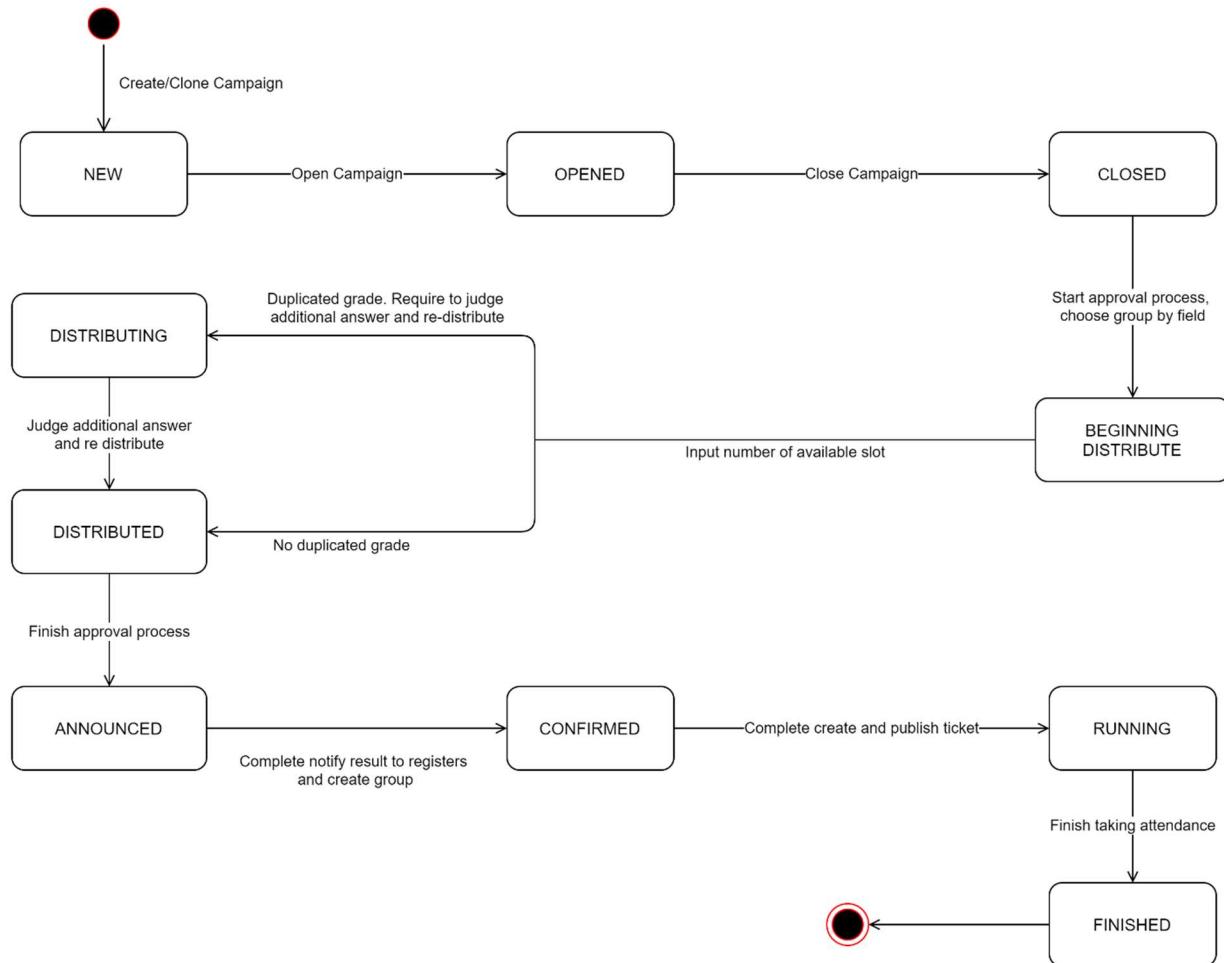


Figure 97 - Campaign Status

No.	State	Description
1	NEW	The status of a campaign when it has just been initialized in the system but is not made public to registrants.
2	OPENED	The status of a campaign after it is made public to registrants. It changes from 'NEW'.
3	CLOSED	The status of a campaign after the registration has been closed. At this time, the registrant cannot register, update or cancel the registration information. It changes from 'CLOSED'.
4	BEGINNING DISTRIBUTE	The status of a campaign after the manager chooses group-by fields to distribute the campaign. It changes from 'CLOSED'.

5	DISTRIBUTING	The status of a campaign after the manager inputs available quantity for each criteria and there are duplicated mark of registrants after the system automatically approves registrants. It changes from 'BEGINNING DISTRIBUTE'.
6	DISTRIBUTED	The status of a campaign after duplicated cases have been resolved by grading extra mark. It changes from 'BEGINNING DISTRIBUTE' or 'DISTRIBUTING'.
7	ANNOUNCED	The status of a campaign after a campaign has been complete approval process, and in the process of notifying approval result to registrants. It changes from 'DISTRIBUTED'.
8	CONFIRMED	The status of a campaign when the manager is creating ticket. It changes from 'ANNOUNCED'.
9	RUNNING	The status of a campaign after tickets are published. It changes from 'CONFIRMED'.
10	FINISHED	The status of a campaign when it completes all process. It changes from 'RUNNING'.

Table 60 - <State Diagram> Campaign Status

### 3.8. User's Resource Status

**Summary:** This diagram shows status of information of registrants after one join the system.

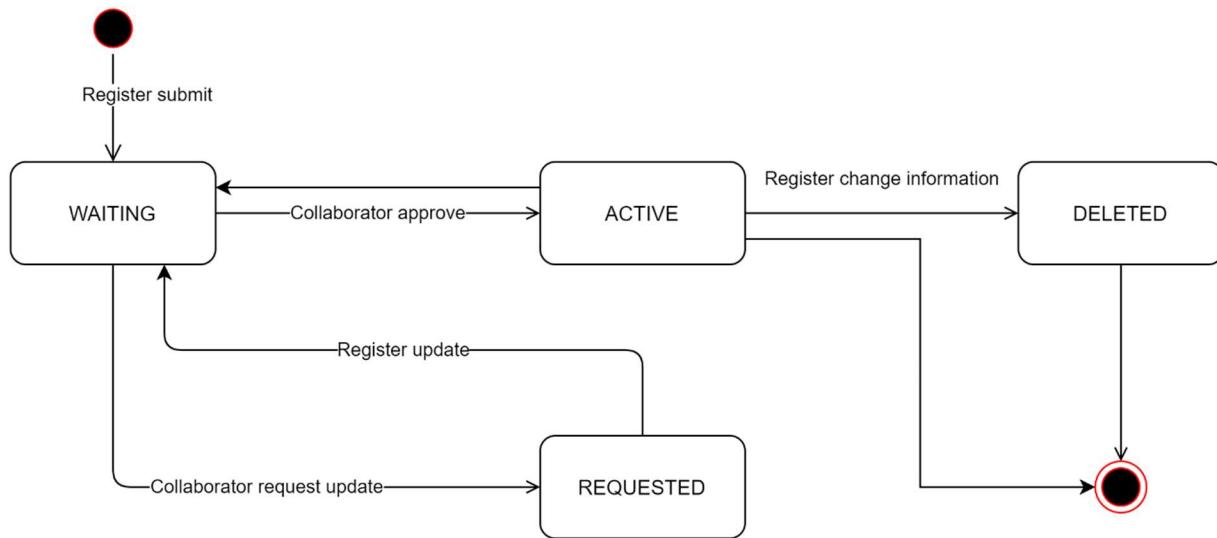


Figure 98 - User's Resource Status

No.	State	Description
1	WAITING	The resource has just been uploaded to the system. The managers have not verified it yet. It may change from 'REQUESTED' or it is an initial state of the resource.
2	ACTIVE	The resource has been verified. It changes from 'WAITING'.
3	DELETED	The resource has been deleted by its owner. It changes from 'ACTIVE'.

4	REQUESTED	The resource is invalid. The manager requests the registrant to update. It changes from 'WAITING'.
---	-----------	--

Table 61 - <State Diagram> User's Resource Status

### 3.9. User registers to Campaign Status

*Summary:* This diagram shows status of a registration of a student after one register to a campaign.

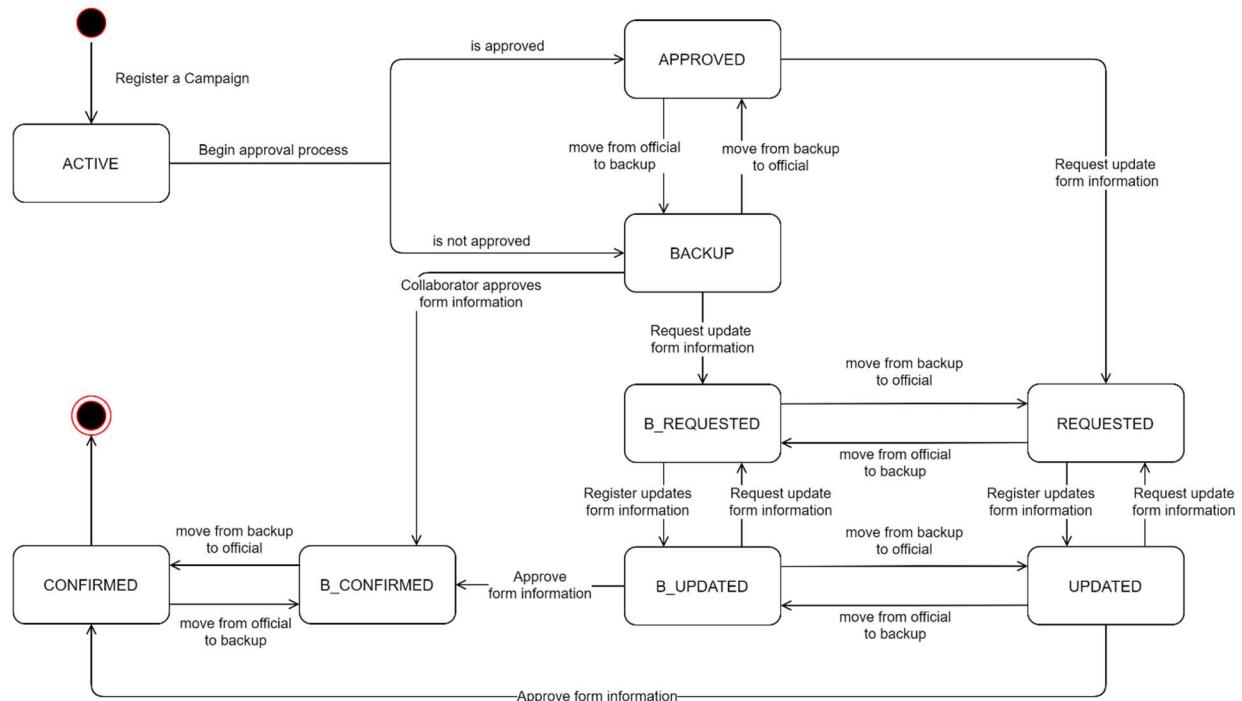


Figure 99 - User registers to Campaign Status

No.	State	Description
1	ACTIVE	The registration has been recorded into the system. It is initial state of the registration.
2	APPROVED	The registration has been approved by the automatic approval. It changes from 'ACTIVE'.
3	BACKUP	The registration is not approved. It changes from 'ACTIVE'.
4	REQUESTED	The registration's form's information is invalid or missing. The registrant needs to update it. The registrant has been approved. It changes from 'APPROVED' OR 'B_REQUESTED'.
5	B_REQUESTED	The same as 'REQUESTED' but the registrant has not been approved. It changes from 'BACKUP' or 'B_REQUESTED'.
6	UPDATED	The registration's form's information has been updated by its owner. It changes from 'REQUESTED' or 'B_UPDATED'.
7	B_UPDATED	The same as 'UPDATED' but the registrant has not been approved. It changes from 'B_REQUESTED' or 'UPDATED'.

8	CONFIRMED	The registration has been confirmed by the managers. It changes from 'UPDATED' or 'B_CONFIRMED'.
9	B_CONFIRMED	The same as 'UPDATED' but the registrant has not been approved by the system. It changes from 'BACKUP' or 'CONFIRMED'.

*Table 62 - <State Diagram> User register Campaign Status*

## 4. Data & Database Design

### 4.1. Table Relationships

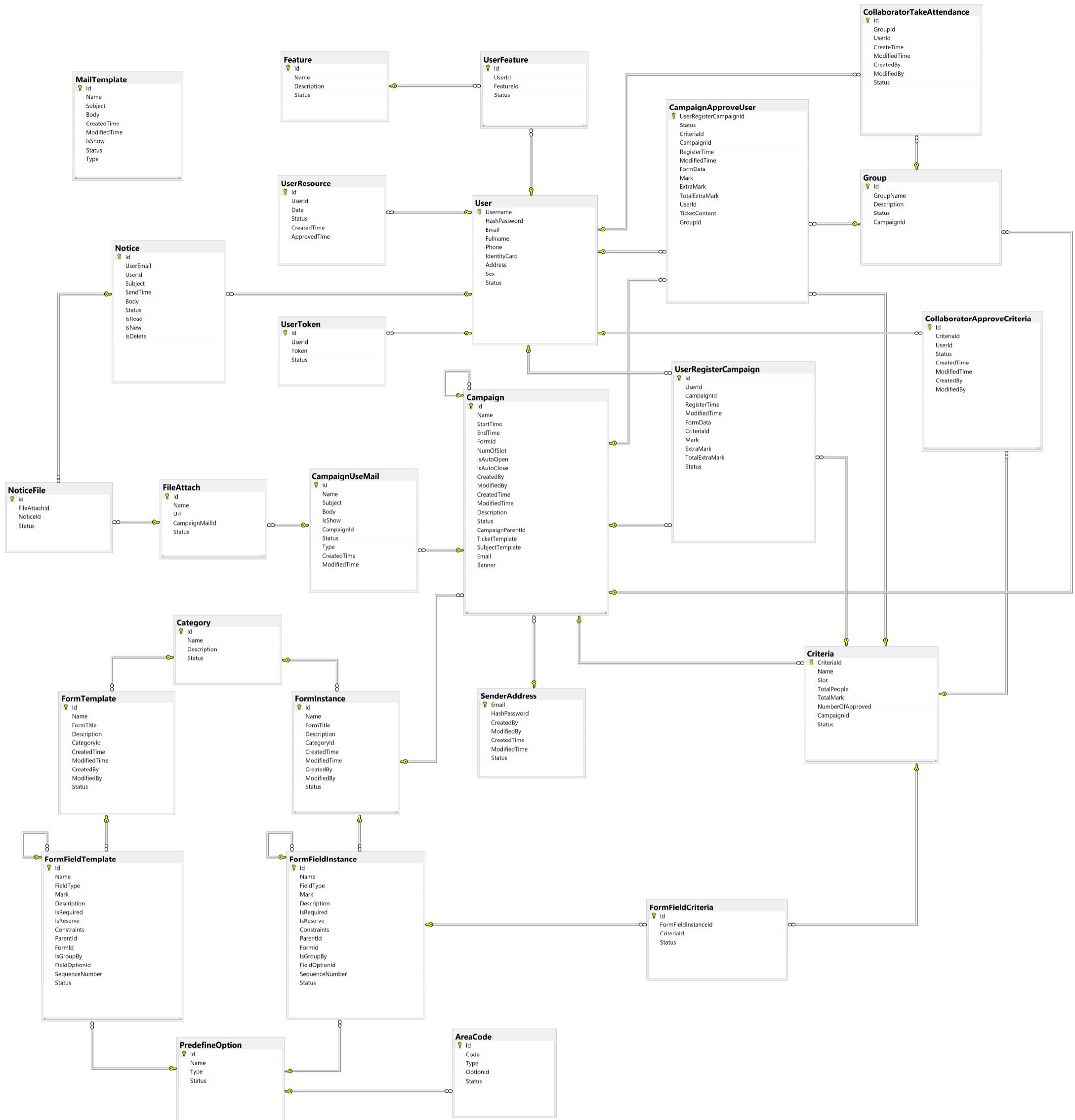


Figure 100 - Physical Diagram

## 4.2. Data dictionary

### User table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
Username	varchar	52	Yes	Yes	PK	Unique identifier of user
HashPassword	varchar	102	No	No		Hash password of user
Email	varchar	52	No	No		Email of user
Fullscreen	nvarchar	102	No	No		Full name of user
Phone	varchar	17	No	No		Phone of user
IdentityCard	varchar	17	No	No		Identity card of user
Address	nvarchar	202	No	No		Address of user
Sex	bit		No	No		Gender of user
Status	nvarchar	20	No	No		Status of user

Table 63 - <Data dictionary> User

### UserToken table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
Id	varchar	52	Yes	Yes	PK	Unique identifier of user token
UserId	varchar	52	No	Yes	FK	Unique identifier of user
Token	varchar	8000	No	Yes		Token detail
Status	nvarchar	22	No	No		Status of user token

Table 64 - <Data dictionary> UserToken

### UserResource table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
Id	varchar	52	Yes	Yes	PK	Unique identifier of resource
UserId	varchar	52	No	Yes	FK	Unique identifier of user
Data	nvarchar	8000	No	No		Data of user resource
CreatedTime	datetime	8	No	No		Created time of user resource
ApprovedTime	datetime	8	No	No		Approved time of user resource
Status	nvarchar	22	No	No		Status of user resource

Table 65 - <Data dictionary> UserResource

### UserFeature table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
Id	int	4	Yes	Yes	PK	Unique identifier of user feature
UserId	varchar	52	No	Yes	FK	Unique identifier of user

<b>FeatureId</b>	int	4	No	Yes	FK	Unique identifier of feature
<b>Status</b>	nvarchar	22	No	No		Status of user feature

Table 66 - <Data dictionary> UserFeature

### Feature table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	int	4	Yes	Yes	PK	Unique identifier of feature
<b>Name</b>	nvarchar	52	No	No		Name of feature
<b>Description</b>	nvarchar	202	No	No		Description of feature
<b>Status</b>	nvarchar	22	No	No		Status of feature

Table 67 - <Data dictionary> Feature

### Notice table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	varchar	52	Yes	Yes	PK	Unique identifier of notice
<b>UserEmail</b>	varchar	102	No	Yes		Email of user
<b>UserId</b>	varchar	52	No	Yes	FK	Unique identifier of user
<b>Name</b>	nvarchar	202	No	No		Name of notice
<b>Subject</b>	nvarchar	202	No	No		Subject of notice
<b>SendTime</b>	datetime	8	No	Yes		Send time of notice
<b>body</b>	nvarchar	8000	No	No		Body of notice
<b>IsRead</b>	bit		No	No		Indicate that notice is read or not
<b>Status</b>	varchar	22	No	No		Status of notice
<b>isNew</b>	bit		No	No		Indicate that notice is new or not
<b>IsDelete</b>	bit		No	No		Indicate that notice is deleted or not

Table 68 - <Data dictionary> Notice

### NoticeFile table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	int	4	Yes	Yes	PK	Unique identifier of notice file
<b>FileAttachId</b>	int	4	No	Yes	FK	Unique identifier of file attach
<b>NoticeId</b>	varchar	52	No	Yes	FK	Unique identifier of notice
<b>Status</b>	nvarchar	52	No	No		Status of notice file

Table 69 - <Data dictionary> NoticeFile

### FileAttach table

Field name	Type	Size (bytes)	Unique	Not Null	PK/ FK	Notes

<b>Id</b>	int	4	Yes	Yes	PK	Unique identifier of file attach
<b>Name</b>	nvarchar	102	No	No		Name of file attach
<b>Url</b>	varchar	8000	No	No		Url of file attach
<b>Status</b>	nvarchar	22	No	No		Status of file attach
<b>CampaignUse MailId</b>	varchar	52	No	No	FK	Unique identifier of mail template

Table 70 - <Data dictionary> FileAttach

### UserRegisterCampaign table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	varchar	52	Yes	Yes	PK	Unique identifier of registrant
<b>UserId</b>	varchar	52	No	Yes	FK	Unique identifier of user
<b>CampaignId</b>	varchar	52	No	Yes	FK	Unique identifier of campaign
<b>RegisterTime</b>	datetime	8	No	yes		Registration time
<b>ModifiedTime</b>	datetime	8	No	No		Last modify time
<b>FormData</b>	nvarchar	8000	No	No		Form registration
<b>CriteriaId</b>	int	4	No	No	FK	Unique identifier of criteria
<b>Mark</b>	int	4	No	No		Mark of registrant
<b>ExtraMark</b>	nvarchar	8000	No	No		Extra mark detail
<b>TotalExtraMark</b>	int	4	No	No		Extra mark of user
<b>Status</b>	nvarchar	22	No	No		Status of registrant

Table 71 - <Data dictionary> UserRegisterCampaign

### Campaign table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	varchar	52	Yes	Yes	PK	Unique identifier of campaign
<b>Name</b>	nvarchar	102	No	No		Name of campaign
<b>StartTime</b>	datetime	8	No	No		Start time of campaign
<b>EndTime</b>	datetime	8	No	No		End time of campaign
<b>FormId</b>	varchar	52	No	Yes	FK	Unique identifier of form
<b>NumOfSlot</b>	int		No	No		Number of slot of campaign
<b>IsAutoOpen</b>	bit		No	No		Indicate that campaign is opened automatically or not
<b>IsAutoClose</b>	bit		No	No		Indicate that campaign is closed automatically or not

<b>CreatedBy</b>	varchar	52	No	Yes		Username of user create campaign
<b>ModifiedBy</b>	varchar	52	No	Yes		Username of user modify campaign
<b>CreatedTime</b>	datetime	8	No	Yes		Create time of campaign
<b>ModifiedTime</b>	datetime	8	No	Yes		Last modify time of campaign
<b>Description</b>	nvarchar	8000	No	No		Description of campaign
<b>Status</b>	varchar	22	No	No		Status of campaign
<b>CampaignParentId</b>	varchar	52	No	No	FK	Unique identifier of parent campaign
<b>TicketTemplate</b>	varchar	8000	No	No		Ticket template of campaign
<b>SubjectTemplate</b>	varchar	202	No	No		Subject template of campaign
<b>Email</b>	varchar	52	No	No		Email used for send mail to user in campaign
<b>Banner</b>	varchar	8000	No	No		Banner url of campaign

Table 72 - <Data dictionary> Campaign

### CampaignUseMail table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	varchar	52	Yes	Yes	PK	Unique identifier of mail
<b>Name</b>	nvarchar	52	No	No		Name of mail
<b>Subject</b>	nvarchar	202	No	No		Subject of mail
<b>Body</b>	nvarchar	8000	No	No		Body of mail
<b>CreatedTime</b>	datetime	8	No	No		Created time of mail
<b>ModifiedTime</b>	datetime	8	No	No		Last modified time of mail
<b>IsShow</b>	bit		No	No		Indicate that mail is show or not
<b>CampaignId</b>	varchar	52	No	No	FK	Unique identifier of campaign use this mail
<b>Status</b>	varchar	22	No	No		Status of mail
<b>Type</b>	varchar	52	No	No		Type of mail

Table 73 - <Data dictionary> CampaignUseMail

### FormInstance & FormTemplate table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	varchar	52	Yes	Yes	PK	Unique identifier of form
<b>Name</b>	nvarchar	102	No	No		Name of form
<b>FormTitle</b>	nvarchar	102	No	No		Title of form
<b>Description</b>	nvarchar	8000	No	No		Description of form

<b>CategoryId</b>	varchar	52	No	Yes	FK	Unique identifier category
<b>CreatedTime</b>	datetime	8	No	No		Created time of form
<b>ModifiedTime</b>	datetime	8	No	No		Last modify time of form
<b>CreatedBy</b>	varchar	52	No	No		Username of user create this form
<b>ModifiedBy</b>	varchar	52	No	No		Username of user modify this form
<b>Status</b>	nvarchar	22	No	No		Status of form field

Table 74 - <Data dictionary> FormInstance & FormTemplate

### FormFieldInstance & FormFieldTemplate table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	varchar	52	Yes	Yes	PK	Unique identifier of form field
<b>Name</b>	nvarchar	202	No	No		Name of form field
<b>FieldType</b>	varchar	52	No	No		Type of form field
<b>Mark</b>	int	4	No	No		Mark of form field
<b>Description</b>	nvarchar	8000	No	No		Description of form field
<b>IsRequired</b>	bit		No	No		Indicate that form field is required or not
<b>IsReserve</b>	bit		No	No		Indicate that form field is use to process extra mark or not
<b>Constraints</b>	varchar	8000	No	No		Constraints expression of form field
<b>ParentId</b>	varchar	52	No	No	FK	Unique identifier of parent form field
<b>FormId</b>	varchar	52	No	Yes	FK	Unique identifier of form
<b>IsGroupBy</b>	bit		No	No		Indicate that form field used to group by or not
<b>FieldOptionId</b>	int	4	No	No		Unique identifier of Option
<b>SequenceNumber</b>	int	4	No	No		Sequence number of form field
<b>Status</b>	nvarchar	22	No	No		Status of form field

Table 75 - <Data dictionary> FormFieldInstance & FormFieldTemplate

### PredefineOption table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	varchar	52	Yes	Yes	PK	Unique identifier of predefine option
<b>Name</b>	nvarchar	102	No	No		Name of option

Type	nvarchar	102	No	No		Type of option
Status	nvarchar	22	No	No		Status of option

Table 76 - <Data dictionary> PredefineOption

### AreaCode table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	int	4	Yes	Yes	PK	Unique identifier of area
<b>Code</b>	varchar	22	No	No		Code of area
<b>Type</b>	varchar	52	No	No		Type of area
<b>OptionId</b>	int	4	No	No	FK	Unique identifier of predefined option
<b>Status</b>	varchar	22	No	No		Status of area

Table 77 - <Data dictionary> AreaCode

### Category table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	int	4	Yes	Yes	PK	Unique identifier of category
<b>Name</b>	nvarchar	52	No	No		Name of category
<b>Description</b>	nvarchar	8000	No	No		Description of category
<b>Status</b>	nvarchar	22	No	No		Status of category

Table 78 - <Data dictionary> Category

### SenderAddress table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Email</b>	varchar	52	Yes	Yes	PK	Unique identifier of sender address
<b>HashPassword</b>	varchar	102	No	No		Hash password of sender address
<b>CreatedBy</b>	varchar	52	No	No		Username of user create this sender address
<b>ModifiedBy</b>	varchar	52	No	No		Username of user modify this sender address
<b>CreatedTime</b>	datetime	8	No	No		Created time of sender address
<b>ModifiedTime</b>	datetime	8	No	No		Last modified time of sender address
<b>Status</b>	nvarchar	22	No	No		Status of sender address

Table 79 - <Data dictionary> Sender Address

### CampaignApproveUser table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
UserRegisterCampaign	varchar	52	Yes	Yes	PK	Unique identifier of registration
Status	varchar	22	No	No		Status of registrant
CriteriaId	int	4	No	No	FK	Unique identifier of criteria
CampaignId	varchar	52	No	Yes	FK	Unique identifier of campaign
RegisterTime	datetime	8	No	No		Registration time of registrant
ModifiedTime	datetime	8	No	No		Last modified time of registrant
FormDate	varchar	8000	No	No		Form data of registrant
Mark	int		No	No		Mark of registrant
ExtraMark	varchar	8000	No	No		Extra mark detail of registrant
TotalExtraMark	int	4	No	No		Extra mark of registrant
UserId	varchar	52	No	No		Username of registrant
TicketContent	varchar	8000	No	No		Ticket content of registrant
GroupId	varchar	52	No	No	FK	Unique identifier of group

Table 80 - <Data dictionary> CampaignApproveUser

### Criteria table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
CriteriaId	int	4	Yes	Yes	PK	Unique identifier of criteria
Name	nvarchar	202	No	No		Name of criteria
Slot	int	4	No	No		Slot of criteria
TotalPeople	int	4	No	No		Total people of criteria
TotalMark	int	4	No	No		Total mark of people in criteria
NumberOfApproved	int	4	No	No		Number of people who have been approved in criteria
CampaignId	varchar	52	No	Yes	FK	Unique identifier of campaign
Status	nvarchar	22	No	No		Status of criteria

Table 81 - <Data dictionary> Criteria

### FormFieldCriteria table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	int	4	Yes	Yes	PK	Unique identifier
<b>FormFieldInstanceId</b>	varchar	52	No	Yes	FK	Unique identifier of form field
<b>CriteriaId</b>	int	4	No	Yes	FK	Unique identifier of criteria
<b>Status</b>	nvarchar	22	No	No		Status of relation between form field and criteria

Table 82 - <Data dictionary> FormFieldCriteria

### Group table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	varchar	52	Yes	Yes	PK	Unique identifier of group
<b>GroupName</b>	nvarchar	102	No	No		Name of group
<b>Description</b>	nvarchar	8000	No	No		Description of group
<b>CampaignId</b>	varchar	52	No	Yes	FK	Unique identifier campaign
<b>Status</b>	nvarchar	22	No	No		Status of form group

Table 83 - <Data dictionary> Group

### CollaboratorApproveCriteria table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	int	4	Yes	Yes	PK	Unique identifier of task
<b>CriteriaId</b>	int	4	No	Yes	FK	Unique identifier of criteria
<b>UserId</b>	varchar	52	No	No	FK	Unique identifier of user
<b>Status</b>	nvarchar	22	No	No		Status of task
<b>CreatedTime</b>	datetime		No	No		Created time of task
<b>ModifiedTime</b>	datetime		No	No		Modified time of task
<b>CreatedBy</b>	varchar	52	No	No		Username of user assign this task
<b>ModifiedBy</b>	varchar	52	No	No		Username of user modify this task

Table 84 - <Data dictionary> CollaboratorApproveCriteria

### CollaboratorTakeAttendance table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	int	4	Yes	Yes	PK	Unique identifier of task
<b>GroupId</b>	varchar	52	No	No	FK	Unique identifier of group
<b>UserId</b>	varchar	52	No	No	FK	Unique identifier of user
<b>Status</b>	nvarchar	22	No	No		Status of task
<b>CreatedTime</b>	datetime	8	No	No		Created time of task
<b>ModifiedTime</b>	datetime	8	No	No		Modified time of task
<b>CreatedBy</b>	varchar	52	No	No		Username of user assign this task
<b>ModifiedBy</b>	varchar	52	No	No		Username of user modify this task

Table 85 - <Data dictionary> CollaboratorTakeAttendance

### MailTemplate table

Field name	Type	Size (bytes)	Unique	Not Null	PK / FK	Notes
<b>Id</b>	varchar	52	Yes	Yes	PK	Unique identifier of mail
<b>Name</b>	nvarchar	52	No	No		Name of mail
<b>Subject</b>	nvarchar	202	No	No		Subject of mail
<b>Body</b>	nvarchar	8000	No	No		Body of mail
<b>CreatedTime</b>	datetime	8	No	No		Created time of mail
<b>ModifiedTime</b>	datetime	8	No	No		Last modified time of mail
<b>IsShow</b>	bit		No	No		Indicate that mail is show or not
<b>Status</b>	varchar	22	No	No		Status of mail
<b>Type</b>	varchar	52	No	No		Type of mail

Table 86 - <Data dictionary> MailTemplate

## 5. Algorithms

### 5.1. Calculate mark

#### 5.1.1. Summary

This algorithm describes the process to calculate mark depends on the information that the registrant provides base on the form when one registers to a campaign. The calculated mark is used in automatic approval process.

Input: form registration information.

Output: a mark for the registration.

### **5.1.2. Define problem**

A campaign can bring up to thousand registrants so we need to filter them. The manager can set a mark on each question in the registration form. After registering, each registrant has a score. The list of registrants will be sorted by mark in descending order. By this way, the manager can handle it easily.

There are 2 type of questions:

- Normal questions are used for calculating mark.
- Extra questions are considered by the manager when registrants get the same mark.

Because the extra question is a text or paragraph type, the system can not handle it. Normal question is divided into a scale, so we need an algorithm to calculate it.

### **5.1.3. Solution**

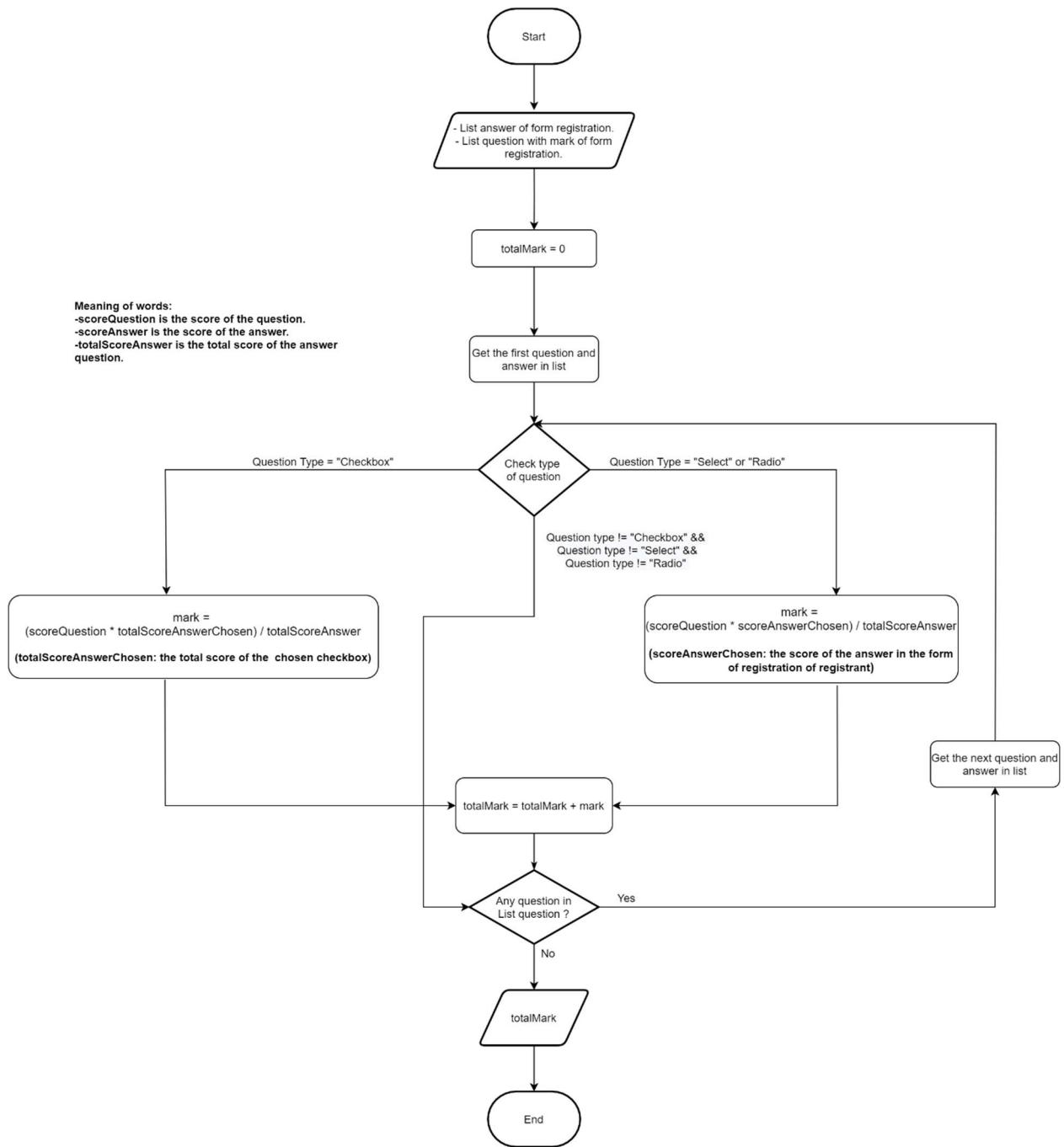


Figure 101 – Flowchart: Calculate mark algorithm

Complexity:  $O(n)$

## 5.2. Approve registrants automatically

### 5.2.1. Summary

This algorithm describes how the system process automatic approve registrants in a criteria.

### **5.2.2. Define problem**

The manager wants to limit the number of slots in criteria. So, the system supports the partial automatic distribution. The system will approve registrants based on their score. The registrants whose high score will be given priority approval in criteria.

When the campaign has over hundreds of registrants, the approval wastes a lot of time and human resources. In the case of the registrant having the same score, it's gonna be hard for the manager. So, we researched a method to overcome this problem.

### **5.2.3. Solution**

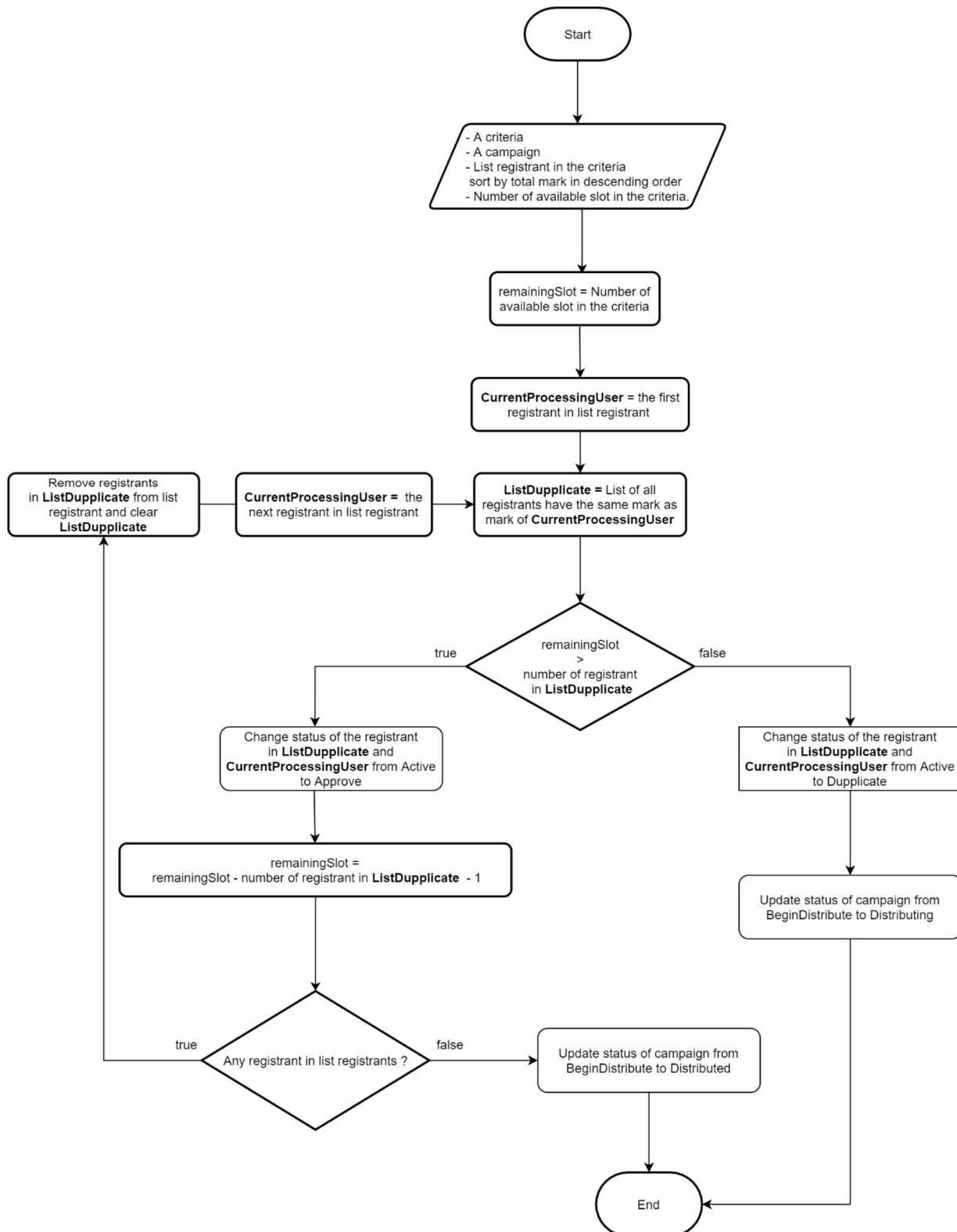


Figure 102 – Flowchart: Approve registrants automatically algorithm

Complexity:  $O(n)$

## **5.3. Distribute registrants into groups**

### **5.3.1. Summary**

When creating a group, the manager needs to choose criterias and input quantity of each criteria that one needs to distribute into the group. This algorithm describes how the system distributes registrants into groups base on list criterias and their quantity.

### **5.3.2. Define problem**

When the campaign has over hundreds of registrants, the approval wastes a lot of time and human resources. In the case of the registrant having the same score, it's gonna be hard for the manager. So, we researched a method to overcome this problem.

When the campaign has over 500 candidates, it would be hard to assign candidates to groups. The distribution wastes a lot of time and human resources. So, we researched a method to overcome this problem.

### **5.3.3. Solution**

The system will support this automatically. Manager just fills in information then the system will base on criterias and slots to assign candidates. The list of candidates is sorted in ascending order by full name before assigning. By this way, the manager can distribute easily and save a lot of time.

